

31-10-2025

TPI PROGRAMACIÓN

Gestión de Datos de Países en Python:
filtros, ordenamientos y estadísticas



python-docx
GALLARDO LUCAS / TORRES SHIRLEY

INTRODUCCION

El presente Trabajo Práctico Integrador (TPI) tiene como objetivo principal diseñar e implementar una aplicación en Python capaz de gestionar, procesar y analizar eficientemente un conjunto de datos geográficos y demográficos de diversos países, obtenidos a través de archivos CSV. Este proyecto no solo sirve como una plataforma para aplicar conceptos fundamentales de programación orientada a datos, sino que también consolida el conocimiento sobre el uso de estructuras de datos avanzadas y la programación modular.

La solución desarrollada se basa en la integración de herramientas y paradigmas esenciales de Python:

Estructuras de Datos: Se emplean Listas para contener el conjunto total de países, y Diccionarios para modelar la información de cada país individualmente (nombre, población, superficie, continente), facilitando la manipulación clave-valor.

Programación Modular: Se utiliza el concepto de Funciones para segmentar el código, asignando una única responsabilidad a cada bloque (lectura de CSV, filtrado, ordenamiento, cálculo de estadísticas). Este enfoque garantiza la legibilidad, mantenibilidad y escalabilidad del código.

Análisis de Datos: El programa incluye funcionalidades para aplicar Condicionales y Ordenamientos que permiten a los usuarios filtrar y clasificar los datos según diversos criterios, así como calcular Estadísticas Básicas (promedios, totales y conteos por categoría) para extraer valor de la información.

El desarrollo sigue un flujo de operaciones estructurado que comienza con la Carga de Datos Iniciales desde un archivo CSV. A partir de allí, el usuario interactúa mediante un Menú Principal que permite: Filtrar Países (por continente o métricas), Ordenar Países (por población o superficie), y Mostrar Estadísticas relevantes. Este informe teórico detalla los conceptos aplicados y presenta un esquema lógico de las operaciones, demostrando la aplicación práctica de cada elemento fundamental de la programación.

En síntesis, este TPI representa un ejercicio integral en el desarrollo de software, abarcando desde la correcta interpretación de datos tabulares (CSV) y su estructuración en Python, hasta la creación de una interfaz funcional y modular para su gestión y análisis.

Trabajo Práctico Integrador (TPI): Gestión de Datos de Países en Python1.

1. INFORME TEÓRICO: Conceptos aplicados

LISTAS

Las listas son colecciones ordenadas y mutables (modificables) de elementos. En Python, pueden contener elementos de diferentes tipos de datos (como *strings*, números, diccionarios, u otras listas). Son ideales para almacenar el conjunto de países que leerás del archivo CSV, donde cada elemento de la lista podría ser un diccionario o una lista que represente un país.

Ejemplo:

```
países = ["Argentina", "Brasil", "Japón"]
```

En este trabajo, se utilizarán listas para guardar múltiples registros de países, donde cada elemento será un diccionario con la información de un país.

DICCIONARIOS

Los diccionarios son colecciones no ordenadas de elementos que se almacenan como pares clave-valor. Las claves deben ser únicas e inmutables (generalmente *strings* o números). Son perfectos para representar un único país, donde las claves serán los campos de datos (como "Nombre", "Población", "Superficie", "Continente") y los valores serán los datos correspondientes.

Ejemplo:

```
país = {  
    "nombre": "Argentina",  
    "poblacion": 45376763,  
    "superficie": 2780400,  
    "continente": "América"  
}
```

Esto permite representar fácilmente cada fila del CSV como un diccionario.

FUNCIONES

Las funciones son bloques de código reutilizables diseñados para realizar una tarea específica. Promueven la modularización, haciendo el código más organizado, legible y fácil de mantener (cumpliendo con la consigna de "una función = una responsabilidad"). Necesitarás funciones para leer el archivo, filtrar países, ordenar, calcular estadísticas y mostrar el menú.

Cada función tendrá una responsabilidad específica (por ejemplo: leer el archivo, filtrar países, calcular promedios, etc.).

Ejemplo:

```
def leer_csv(ruta_archivo):
```

```
# Lee los datos del CSV y devuelve una lista de diccionarios
pass
```

CONDICIONALES

Las estructuras condicionales (if, elif, else) permiten que el programa tome decisiones y ejecute diferentes bloques de código basándose en el cumplimiento de ciertas condiciones (expresiones booleanas). Se usarán para el menú de opciones, la validación de entradas del usuario y los filtros sobre los datos (ej. "si el continente es 'América', incluir el país").

Ejemplo:

```
if pais["continente"] == "América":
    print(pais["nombre"])
```

Se usarán para aplicar filtros y validaciones (por ejemplo, al buscar países o verificar datos ingresados por el usuario).

ORDENAMIENTOS

Los ordenamientos son algoritmos que reorganizan los elementos de una lista según un criterio específico (ej. ascendente o descendente). En este TPI, ordenarás la lista de países según campos como "Población" o "Superficie". Python facilita esto con el método `sort()` o la función `sorted()`, a menudo utilizando una función *lambda* para especificar la clave de ordenamiento en los diccionarios.

Ejemplo:

```
países_ordenados = sorted(paises, key=lambda p: p["poblacion"], reverse=True)
```

ESTADÍSTICAS BÁSICAS

Las estadísticas básicas implican calcular indicadores clave sobre un conjunto de datos numéricos. Para este TPI, las estadísticas relevantes serán:

1. Suma/Total: Suma de todas las poblaciones o superficies.
2. Promedio (Media): Suma total de una métrica dividida por el número de países.
3. Máximo/Mínimo: Encontrar el país con la mayor o menor población/superficie.
4. Conteo por Categoría: Contar cuántos países pertenecen a cada continente.

Ejemplo:

```
promedio = sum(p["poblacion"] for p in paises) / len(paises)
```

ARCHIVOS CSV

Un archivo CSV (*Comma-Separated Values*) es un formato de texto simple donde los datos se separan por comas (o a veces por otros delimitadores). Es una forma común y sencilla de almacenar datos tabulares. Necesitarás funciones para leer este archivo,

interpretando la primera línea como encabezados (las claves del diccionario) y cada línea subsiguiente como un registro de país. En Python, esto se maneja de forma eficiente con el módulo estándar csv.

Ejemplo:

```
import csv
def leer_csv(ruta):
    paises = []
    with open(ruta, newline="", encoding='utf-8') as archivo:
        lector = csv.DictReader(archivo)
        for fila in lector:
            fila["poblacion"] = int(fila["poblacion"])
            fila["superficie"] = int(fila["superficie"])
            paises.append(fila)
    return paises
```

2. FLUJO DE OPERACIONES PRINCIPALES (Esquema)

El diagrama de flujo describe la secuencia lógica de cómo el usuario interactúa con la aplicación, desde el inicio hasta el fin.

1. Inicio

2. Cargar Datos Iniciales (Proceso)

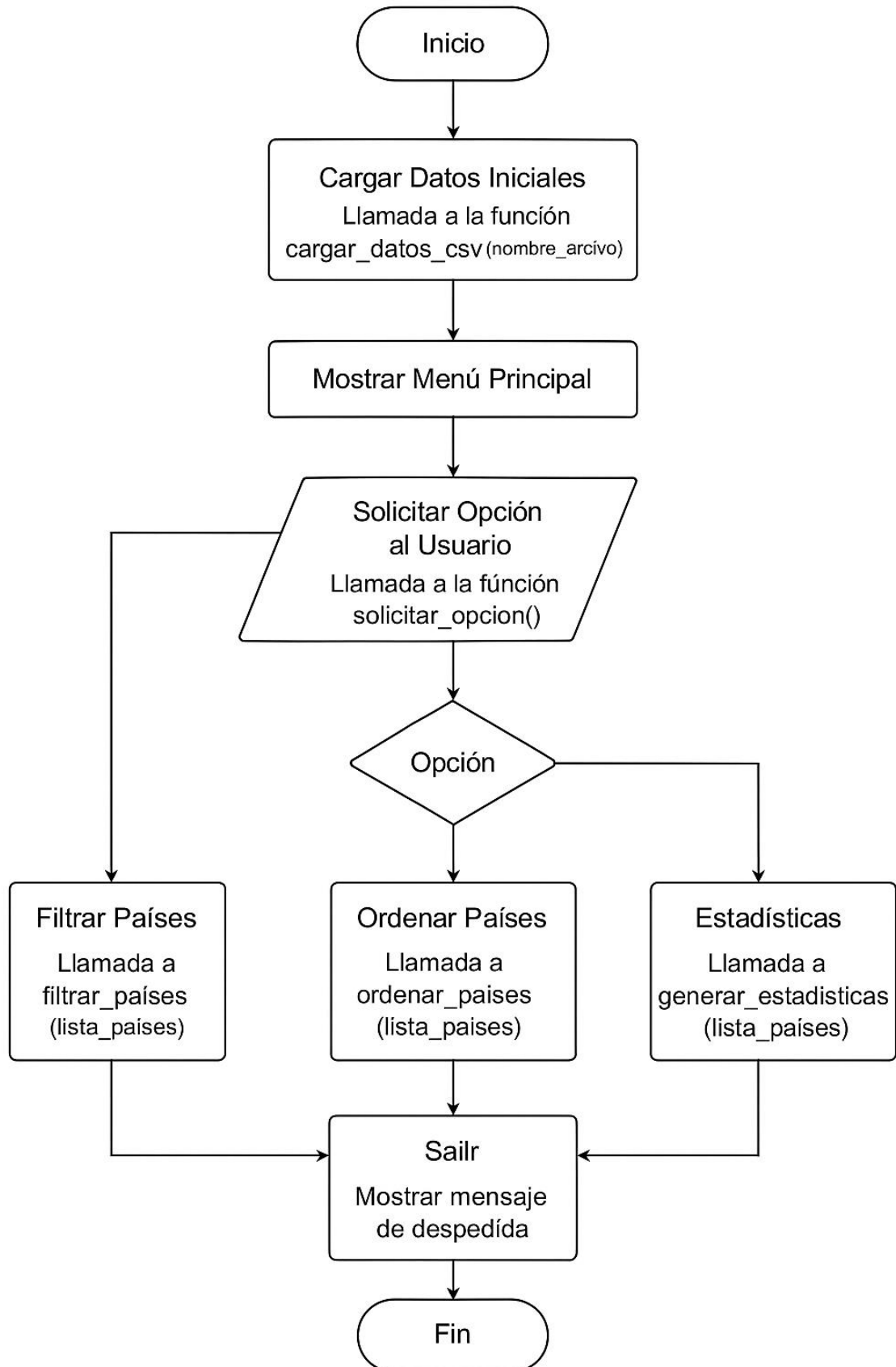
- Llamada a la función `cargar_datos_csv(nombre_archivo)`
- Si la carga falla (archivo no encontrado o error de formato): mostrar error y volver a intentar o salir.
- Guardar la lista de diccionarios de países en una variable principal.

3. Mostrar Menú Principal (Proceso)

- Presentar las opciones disponibles al usuario (Filtrar, Ordenar, Estadísticas, Salir).

4. Solicitar Opción al Usuario (Entrada/Salida)

- Llamada a la función `solicitar_opcion()`
- Validación: Verificar que la entrada sea un número entero y esté dentro del rango de opciones. Si no es válida, volver a solicitar.



5. Decisión (Opción) (Condicional)

- A. Si la opción es "Filtrar Países":
 - Llamada a `filtrar_paises(lista_paises)`.
 - Mostrar submenú de filtros (por Continente, por Población Mínima, etc.).
 - Aplicar el filtro y mostrar la lista de resultados.
 - → Volver al Menú Principal (Paso 3).
- B. Si la opción es "Ordenar Países":
 - Llamada a `ordenar_paises(lista_paises)`.
 - Solicitar criterio de ordenamiento (Nombre, Población, Superficie) y dirección (Ascendente/Descendente).
 - Aplicar el ordenamiento y mostrar la lista ordenada.
 - → Volver al Menú Principal (Paso 3).
- C. Si la opción es "Estadísticas":
 - Llamada a `generar_estadisticas(lista_paises)`.
 - Mostrar estadísticas (Población total, Superficie promedio, Conteo por continente, etc.).
 - → Volver al Menú Principal (Paso 3).
- D. Si la opción es "Salir":
 - Mostrar mensaje de despedida.
 - → Fin.
- E. Si la opción es Inválida:
 - Mostrar error.
 - → Volver al Menú Principal (Paso 3).

5. Fin

3. APIs, Programación Modular, Manejo de Información de Países y Archivos CSV

Uso de la API REST de API Countries

Para obtener información sobre países, como población, superficie y continente, usamos la **API REST de API Countries**. Esta API proporciona datos estructurados en formato JSON que se pueden procesar fácilmente en Python.

¿Qué es una API?

Una **API (Interfaz de Programación de Aplicaciones)** permite que dos programas se comuniquen mediante un conjunto de reglas y protocolos. Por ejemplo, una aplicación de clima en tu teléfono obtiene datos de un instituto meteorológico a través de una API.

Cómo funcionan las APIs

Las APIs operan bajo un modelo **cliente-servidor**: - **Cliente**: solicita información (ej. app móvil). - **Servidor**: responde con datos (ej. base de datos meteorológica).

En este caso, usamos **REST**: - REST significa *Representational State Transfer*. - Usa funciones como GET, POST, PUT, DELETE para acceder a datos. - Es **sin estado**: el servidor no guarda información del cliente entre solicitudes.

Programación Modular

La **programación modular** consiste en dividir un programa grande en subprogramas independientes llamados **módulos**, cada uno con funciones específicas.

Ventajas

- **Facilidad de uso:** se trabaja con partes más pequeñas del código, lo que simplifica la depuración.
- **Reutilización:** los módulos pueden ser reutilizados en otros proyectos sin rehacer el código.
- **Mantenimiento:** permite que varios desarrolladores trabajen simultáneamente sin conflictos.

Consideraciones

- Definir límites claros para cada módulo.
- Planificar cómo dividir el programa en módulos.
- Establecer comunicación correcta entre módulos para garantizar la funcionalidad del programa completo.

Manejo de Información de Países

La información de países, como **población, superficie y continente**, se obtiene mediante la **API REST de API Countries**, que devuelve datos en formato JSON. En **Python**, se puede usar la biblioteca requests para realizar solicitudes GET y procesar la información obtenida.

Archivos CSV

Un **archivo CSV (Comma Separated Values)** es un formato de texto plano para almacenar datos tabulares. Cada línea representa un registro y los campos se separan por comas. Es muy utilizado por su simplicidad y compatibilidad con hojas de cálculo y bases de datos.

En **Python**, la biblioteca csv permite leer y escribir archivos CSV de manera eficiente. Por ejemplo, se puede guardar información sobre países escribiendo un archivo CSV con los encabezados y valores correspondientes.

Capturas de pantalla de ejecución de ejemplos

Ejemplo de ejecución de la primera opción del menú (Buscar país). Elegimos la opción correspondiente.


```
PS C:\Users\Usuario\Documents\GitHub\IntegradorUTN1> py main.py
4. Mostrar estadísticas:
a) País con mayor y menor población
b) Promedio de población
c) Promedio de superficie
d) Cantidad de países por continente

0. Salir
=====
Ingrese una opcion: 1
```

Buscamos “arg” y nos aparece el resultado de la búsqueda.

```
Ingrese una opcion: 1
Ingrese el nombre del pais: arg
Países 1 - 1
```

Pais	Superficie	Poblacion	Continente
Argentina	2780400.0	45376763	Americas

Ejemplo de la ejecución de la segunda opción del menú (Filtrar países).

Elegimos la opción correspondiente y nos aparece el siguiente menú.

```
Ingrese una opcion: 2

--- Opciones de filtrado (Opción 2) ---
a) Continente
b) Rango de población
c) Rango de superficie
d) Volver al menú principal

Ingrese una subopcion de filtrado: 
```

Al elegir por continente, nos aparece un nuevo menú para filtrar países por los continentes correspondientes.

```
--- Continentes Disponibles ---
- Africa
- Americas
- Antarctic
- Asia
- Europe
- Oceania
-----

Ingrese el nombre del continente a filtrar: 
```

Elegimos “África” y nos aparecen los resultados. Si queremos continuar, simplemente presionamos cualquier botón, excepto los indicados para salir.

```
Resultados del filtro por continente ('Africa'):
Países 1 - 7
```

Pais	Superficie	Poblacion	Continente
Benin	112622.0	12123198	Africa
Ghana	238533.0	31072945	Africa
Ivory Coast	322463.0	26378275	Africa
Republic of the Congo	342000.0	5657000	Africa
Ethiopia	1104300.0	114963583	Africa
Senegal	196722.0	16743930	Africa
Guinea-Bissau	36125.0	1967998	Africa

```
¿Siguiente pagina? (escriba 'no' o 'n' para salir) 
```

Ejemplo de ejecución de la tercera opción (Ordenar).

Elegimos la opción

```
Ingrese una opcion: 3

--- Opciones de ordenamiento (Opción 3) ---
a) Ordenar por nombre
b) Ordenar por poblacion
c) Ordenar por superficie
d) Volver al menú principal

Ingrese una subopcion de ordenamiento: 
```

Elegimos “Ordenar por población” y nos aparecerá el resultado.

Ingrese una subopción de ordenamiento: b

Países 1 - 7

País	Superficie	Poblacion	Continente
Heard Island and McDonald Islands	412.0	0	Antarctic
Bouvet Island	49.0	0	Antarctic
South Georgia	3903.0	30	Antarctic
Pitcairn Islands	47.0	56	Oceania
United States Minor Outlying Islands	34.2	300	Americas
French Southern and Antarctic Lands	7747.0	400	Antarctic
Vatican City	0.44	451	Europe

¿Siguiente pagina? (escriba 'no' o 'n' para salir)

Elegimos la cuarta opción (Mostrar estadísticas).

3. Ordenar países por:
- a) Nombre
 - b) Población
 - c) Superficie (ascendente o descendente)

4. Mostrar estadísticas:
- a) País con mayor y menor población
 - b) Promedio de población
 - c) Promedio de superficie
 - d) Cantidad de países por continente

0. Salir

=====

Ingrese una opcion: 4

Ln 22

Nos aparecerá el siguiente menú y elegiremos qué estadística queremos.

Ingrese una opcion: 4

--- Opciones de estadísticas (Opción 4) ---

- a) País con menor y mayor población
- b) Promedio de población
- c) Promedio de superficie
- d) Cantidad de países por continente
- e) Volver al menú principal

Ingrese una subopción de estadística: a

País con menor población: Heard Island and McDonald Islands (0)

País con mayor población: China (1402112000)

Conclusion

Durante el desarrollo del trabajo, aprendimos a trabajar con **módulos, funciones y estructuras de datos en Python**. También comprendimos la importancia de **planificar el código en equipo**, de **comentar correctamente las funciones** y de **usar APIs externas para obtener información real**. Este proyecto resultó especialmente útil para **entender cómo se consumen las APIs** y cómo integrar los datos obtenidos en aplicaciones propias. Además, nos permitió **practicar e implementar buenas prácticas de programación**, fortaleciendo nuestras habilidades tanto en desarrollo de software como en trabajo colaborativo.

Bibliografia

1. APIs y Requests en Python

<https://aws.amazon.com/es/what-is/api/>

https://www.w3schools.com/python/module_requests.asp

2. Archivos CSV y manejo de datos

https://www.w3schools.com/python/ref_module_csv.asp

3. Programacion modular

<https://www.geeksforgeeks.org/software-engineering/modular-approach-in-programming/>