

INFORME TEÓRICO

SISTEMA DE GESTIÓN DE PAÍSES

Concepto	Definición	Aplicación en Código
Listas	Una estructura de datos fundamental en Python utilizada para almacenar una colección ordenada y mutable de elementos. Permiten acceder a los elementos por índice.	La variable países en <code>datos.py</code> y <code>main.py</code> es una lista. Almacena todos los países cargados del CSV. Por ejemplo, en <code>busqueda.py</code> : <code>resultados = [p for p in países if ...]</code> crea una nueva lista de resultados filtrados.

Concepto	Definición	Aplicación en Código
Diccionarios	Una estructura de datos que almacena datos en formato de pares clave-valor (o <i>mapping</i>). Son mutables y permiten una recuperación de datos muy rápida usando la clave.	Cada país individual es un diccionario . En <code>datos.py</code> , el <code>csv.DictReader</code> carga cada fila del CSV como un diccionario: <code>{'nombre': 'Argentina', 'poblacion': 45376763, ...}</code> . Se accede a los atributos con la clave, ej.: <code>p['poblacion']</code> .

Concepto	Definición	Aplicación en Código
Funciones	Bloques de código reutilizables diseñados para realizar una tarea específica. Promueven la modularidad , haciendo el código más organizado, legible y fácil de mantener.	El sistema está completamente modularizado: <code>cargar_datos()</code> (carga), <code>guardar_paises()</code> (persistencia), <code>mostrar_menu()</code> (UI), <code>buscar_pais()</code> (lógica de búsqueda), etc. Cada archivo (<code>datos.py</code> , <code>busqueda.py</code> , etc.) agrupa funciones relacionadas.

Concepto	Definición	Aplicación en Código
Condicionales	Estructuras de control que permiten ejecutar código diferente dependiendo de si una o más condiciones booleanas son True o False. Incluye if, elif, y else.	* Control de Flujo: En main.py, la estructura match opcion: (equivalente a un conjunto de if/elif en versiones anteriores) dirige la ejecución a la función correcta basada en la opción del menú elegida.
Condicionales de Bucle	* Validación: En busqueda.py y edicion.py, los bucles while True con una condición if break validan que los inputs sean correctos (ej., que la población mínima no sea mayor a la máxima).	

Concepto	Definición	Aplicación en Código
Ordenamientos	Proceso de organizar una colección de datos en una secuencia específica (ascendente o descendente) basada en uno o más criterios (claves).	En busqueda.py (función ordenar_paises_opcion), se utiliza la función sorted() junto con una función lambda para ordenar la lista de diccionarios por nombre, poblacion o superficie, según lo elija el usuario: <code>sorted(paises, key=lambda x: x[clave], reverse=descendente)</code> .

Concepto	Definición	Aplicación en Código
Estadísticas Básicas	Cálculos numéricos que resumen, describen y analizan un conjunto de datos (ej., máximo, mínimo, suma, promedio y recuento).	En visualizacion.py (función <code>mostrar_estadisticas</code>):
Funciones Agregadas	* Se usan las funciones max() y min() para encontrar los países con mayor y menor población/superficie.	
Suma y Promedio	* Se utiliza sum() para calcular el total de población y superficie, y luego se divide entre <code>len(paises)</code> para obtener los promedios .	
Conteo	* Se usa un diccionario (<code>continentes = {}</code>) y un bucle para realizar un conteo por continente , aplicando el patrón de <code>continentes.get(cont, 0) + 1</code> .	

Concepto	Definición	Aplicación en Código
Archivos CSV (Comma Separated Values)	Un formato de archivo simple que almacena datos tabulares (texto simple) donde los valores son separados por comas (o algún otro delimitador). Es crucial para la persistencia de datos.	En datos.py :
<code>csv.DictReader</code>	* Utilizado en <code>cargar_datos()</code> para leer el archivo <code>paises.csv</code> . Convierte automáticamente cada fila en un diccionario de Python, usando la primera fila (los encabezados) como claves.	
<code>csv.DictWriter</code>	* Utilizado en <code>guardar_paises()</code> para escribir los datos. Se le pasan las claves (<code>fieldnames</code>) y escribe la lista de diccionarios de vuelta al archivo, asegurando que el formato se mantenga y que se incluya el encabezado.	
Gestión de Archivos	* El uso de with open(...) garantiza que el archivo se cierre automáticamente, incluso si ocurre un error, previniendo la corrupción de datos.	

Conclusiones Finales

El desarrollo del **Sistema de Gestión de Datos de Países** demuestra una comprensión sólida de la programación modular en Python.

1. **Separación de Responsabilidades:** Al dividir el código en módulos (datos.py, busqueda.py, edicion.py, etc.), se logra una **estructura limpia** donde cada archivo tiene un único propósito (ej., datos.py solo maneja la persistencia).
2. **Integridad de Datos:** El uso de **csv.DictReader/DictWriter** asegura que los datos se lean y escriban en un formato estructurado y consistente. Además, las **validaciones estrictas** en las funciones de edición y adición previenen la introducción de datos incorrectos (ej., números negativos o campos vacíos).
3. **Manejo de Errores:** La inclusión de bloques **try...except** en la carga de datos es crucial, haciendo la aplicación **robusta** frente a la ausencia del archivo o la existencia de filas con formato inválido.
4. **Eficiencia en Memoria:** Los datos se cargan una única vez al inicio en la lista 'países' y se manipulan eficientemente en memoria (filtrado, ordenamiento), lo cual es rápido. La persistencia (guardar_paises) se realiza solo cuando un cambio (Alta/Edición/Eliminación) lo requiere.

-Agregamos las funciones editar y eliminar país ya que lo veíamos necesario para mejorar el sistema.

-Y en la parte de mostrar estadísticas añadimos la muestra de los países con mayor y menor superficie.

-Nos gustó mucho programar este código por su legibilidad y validaciones que hay que realizar, se nos complicó la parte de importar el csv de países pero lo pudimos solucionar.

Fuentes Bibliográficas

- Python Software Foundation. (n.d.). *The Python Standard Library (csv, os, math)*.
- McKinney, W. (2018). *Python for Data Analysis*. O'Reilly Media. (Referencia para estructuras de datos y manejo de archivos).

- Lutz, M. (2013). *Learning Python*. O'Reilly Media. (Referencia para funciones, módulos y estructuras de control).



- □ Verde: Inicio/Fin y confirmaciones
- ● Azul: Procesos principales
- □ Naranja: Decisiones
- ● Rojo: Errores

#EJEMPLO DE EJECUCIÓN OPCIÓN 1

```

=== GESTIÓN DE DATOS DE PAÍSES ===
1. Buscar país por nombre
2. Filtrar por continente
3. Filtrar por rango de población
4. Filtrar por rango de superficie
5. Ordenar países
6. Mostrar estadísticas
7. Agregar país
8. Editar país
9. Eliminar país
0. Salir
Seleccione una opción: 1

---BUSCAR PAÍS---
Ingrese nombre (o parte del nombre) del país: ar

País                Población    Superficie (km²) Continente
-----
Argentina           45,376,763   2,780,400     América
Paraguay            7,656,330    406,752       América
-----

```

#EJEMPLO DE EJECUCIÓN OPCIÓN 2

```
=== GESTIÓN DE DATOS DE PAÍSES ===
1. Buscar país por nombre
2. Filtrar por continente
3. Filtrar por rango de población
4. Filtrar por rango de superficie
5. Ordenar países
6. Mostrar estadísticas
7. Agregar país
8. Editar país
9. Eliminar país
0. Salir
Seleccione una opción: 2

---FILTRAR POR CONTINENTE---
Ingrese continente (América, Europa, Asia, África, Oceanía): américa

País          Población    Superficie (km²) Continente
-----
Argentina     45,376,763   2,780,400    América
Brasil        213,993,437  8,515,767    América
Canadá        38,246,108   9,984,670    América
Paraguay      7,656,330    406,752      América
-----
```

#EJEMPLO DE EJECUCIÓN OPCIÓN 3

```
=== GESTIÓN DE DATOS DE PAÍSES ===
1. Buscar país por nombre
2. Filtrar por continente
3. Filtrar por rango de población
4. Filtrar por rango de superficie
5. Ordenar países
6. Mostrar estadísticas
7. Agregar país
8. Editar país
9. Eliminar país
0. Salir
Seleccione una opción: 3

---FILTRAR POR POBLACIÓN---
Población mínima: 60000000
Población máxima: 40000000
La población mínima no puede ser mayor que la máxima.
Población mínima: █
```

#EJEMPLO DE EJECUCIÓN OPCIÓN 4

=== GESTIÓN DE DATOS DE PAÍSES ===

1. Buscar país por nombre
2. Filtrar por continente
3. Filtrar por rango de población
4. Filtrar por rango de superficie
5. Ordenar países
6. Mostrar estadísticas
7. Agregar país
8. Editar país
9. Eliminar país
0. Salir

Seleccione una opción: 4

---FILTRAR POR SUPERFICIE---

Superficie mínima (km²): 500000

Superficie máxima (km²): 1000000

País	Población	Superficie (km²)	Continente
Francia	67,750,000	551,695	Europa
Nigeria	211,400,708	923,768	África

#EJEMPLO DE EJECUCIÓN OPCIÓN 5

3. Filtrar por rango de población
4. Filtrar por rango de superficie
5. Ordenar países
6. Mostrar estadísticas
7. Agregar país
8. Editar país
9. Eliminar país
0. Salir

Seleccione una opción: 5

---ORDENAR PAÍSES---

Ordenar por:

- a) Nombre
- b) Población
- c) Superficie

Elija (a/b/c): A

¿Ascendente (a) o Descendente (d)? a

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	38,246,108	9,984,670	América
China	1,412,600,000	9,596,961	Asia
Egipto	104,258,327	1,001,450	África
Francia	67,750,000	551,695	Europa
India	1,380,004,385	3,287,263	Asia
Japón	125,800,000	377,975	Asia
Nigeria	211,400,708	923,768	África
Paraguay	7,656,330	406,752	América
Rusia	143,400,000	17,098,246	Asia

#EJEMPLO DE EJECUCIÓN OPCIÓN 6

```
=== GESTIÓN DE DATOS DE PAÍSES ===
1. Buscar país por nombre
2. Filtrar por continente
3. Filtrar por rango de población
4. Filtrar por rango de superficie
5. Ordenar países
6. Mostrar estadísticas
7. Agregar país
8. Editar país
9. Eliminar país
0. Salir
Seleccione una opción: 6

---MOSTRAR ESTADÍSTICAS---

ESTADÍSTICAS GENERALES
- País con mayor población: China (1,412,600,000)
- País con menor población: Paraguay (7,656,330)
- País con mayor superficie: Rusia (17,098,246 km²)
- País con menor superficie: Alemania (357,022 km²)
- Población promedio: 296,870,954
- Superficie promedio: 4,813,384 km²

Cantidad de países por continente:
- América: 4
- Asia: 4
- Europa: 2
- África: 2
- Oceanía: 1
-----
```

#EJEMPLO DE EJECUCIÓN OPCIÓN 7

```
=== GESTIÓN DE DATOS DE PAÍSES ===
1. Buscar país por nombre
2. Filtrar por continente
3. Filtrar por rango de población
4. Filtrar por rango de superficie
5. Ordenar países
6. Mostrar estadísticas
7. Agregar país
8. Editar país
9. Eliminar país
0. Salir
Seleccione una opción: 7

---AGREGAR PAÍS---
Nombre del país: Colombia
Población: 52700000
Superficie (km²): 1141748
Continente (América, Europa, Asia, África, Oceanía): América
País 'Colombia' agregado correctamente.
-----
```


#EJEMPLO DE EJECUCIÓN OPCIÓN 8

```
=== GESTIÓN DE DATOS DE PAÍSES ===
1. Buscar país por nombre
2. Filtrar por continente
3. Filtrar por rango de población
4. Filtrar por rango de superficie
5. Ordenar países
6. Mostrar estadísticas
7. Agregar país
8. Editar país
9. Eliminar país
0. Salir
Seleccione una opción: 8

---EDITAR PAÍS---
Nombre del país a editar: Colombia

Editando: Colombia
Nuevo nombre [Colombia]:
Nueva población [52700000]: 53000000
Nueva superficie (km²) [1141748]:
Nuevo continente [América]:
País actualizado correctamente.
-----
```

#EJEMPLO DE EJECUCIÓN OPCIÓN 9

```
=== GESTIÓN DE DATOS DE PAÍSES ===
1. Buscar país por nombre
2. Filtrar por continente
3. Filtrar por rango de población
4. Filtrar por rango de superficie
5. Ordenar países
6. Mostrar estadísticas
7. Agregar país
8. Editar país
9. Eliminar país
0. Salir
Seleccione una opción: 9

---ELIMINAR PAÍS---
Nombre del país a eliminar:
Este campo no puede estar vacío.
Nombre del país a eliminar: Colombia
¿Está seguro de eliminar 'Colombia'? (s/n): si
País 'Colombia' eliminado correctamente.
-----
```

