

INFORME TEÓRICO — TRABAJO PRÁCTICO INTEGRADOR

Programación 1 – Gestión de Datos de Países en Python

1) LISTAS

Las listas son estructuras de datos ordenadas y mutables, que permiten almacenar múltiples elementos dentro de una misma variable.

Ejemplo:

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

En el proyecto, se usa una lista para almacenar todos los países cargados desde el CSV, donde cada elemento es un diccionario.

2) DICCIONARIOS

Los diccionarios permiten almacenar información utilizando pares clave–valor. Son ideales para representar entidades complejas como países.

Ejemplo:

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

Cada país del dataset se representa como un diccionario y se almacena dentro de una lista general.

3) FUNCIONES

Las funciones permiten modularizar el código y evitar repeticiones. Cada función debe cumplir una responsabilidad específica.

Ejemplo:

```
def buscar_pais(países, nombre):  
  
    return [p for p in países if nombre.lower() in p["nombre"].lower()]
```

En el TPI, se implementan funciones para buscar, filtrar, ordenar, calcular estadísticas y leer el CSV.

4) ESTRUCTURAS CONDICIONALES

Las estructuras condicionales permiten tomar decisiones en función de condiciones lógicas.

Ejemplo:

```
if pais["poblacion"] > 100000000:
```

```
    print("País muy poblado")
```

```
else:
```

```
    print("Población moderada")
```

Estas estructuras se usan para validar entradas, evitar errores y mostrar mensajes adecuados.

5) ORDENAMIENTOS

Python permite ordenar listas mediante `sorted()` y `sort()`.

Ejemplos:

Ordenar países por población ascendente:

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

Ordenar por superficie descendente:

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

6) ESTADÍSTICAS BÁSICAS

El proyecto requiere varios cálculos estadísticos.

Ejemplos:

Promedio de población:

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

País con mayor población:

```
max_pais = max(paises, key=lambda p: p["poblacion"])
```

Cantidad de países por continente:

```
conteo = {}
```

```
for p in paises:
```

```
    conteo[p["continente"]] = conteo.get(p["continente"], 0) + 1
```

7) ARCHIVOS CSV

Un archivo CSV contiene datos separados por comas. Python permite leerlo con el módulo csv.

Ejemplo:

```
import csv
```

```
with open("paises.csv", newline="", encoding="utf-8") as archivo:
```

```
    lector = csv.DictReader(archivo)
```

```
    lista = [dict(row) for row in lector]
```

El CSV debe convertirse en diccionarios y se deben transformar los campos numéricos.

8) FLUJO DEL SISTEMA

1. Leer el archivo CSV.
2. Convertir cada registro en un diccionario.
3. Guardar todos los países en una lista.
4. Mostrar menú al usuario.
5. Permitir búsquedas, filtros, ordenamientos y estadísticas.
6. Validar entradas.
7. Mostrar resultados.
8. Repetir hasta que el usuario decida salir.

CONCLUSIÓN

Este trabajo integrador permite aplicar todos los conceptos fundamentales de Programación 1: estructuras de datos, modularización, manejo de archivos y análisis mediante estadísticas. Representa una experiencia práctica completa de procesamiento de datos reales.

