

Marco Teórico: Fundamentos de Programación y Estructuras de Datos en Python

El proyecto "Gestión de Datos de Países" aplica conceptos fundamentales de la programación estructurada y las estructuras de datos de Python para manipular y analizar información de un conjunto de datos (dataset).

1. Estructuras de Datos Principales

- **Listas (List)** Una Lista es una colección ordenada y mutable de elementos. En Python, se definen con corchetes ([]) y pueden contener elementos de diferentes tipos (heterogéneas).
 - **Aplicación en el proyecto:** La estructura de datos principal para almacenar el conjunto de países es una Lista de Diccionarios. Cada elemento de esta lista representa un país, y la lista completa es el dataset cargado desde el archivo CSV, manipulado y procesado por las funciones del programa.
- **Diccionarios (Dict)** Un Diccionario es una estructura de datos que almacena información en pares clave-valor. Las claves (keys) son únicas y se utilizan para acceder a los valores (values). Los diccionarios son mutables y dinámicos.
 - **Aplicación en el proyecto:** Cada país en el proyecto se modela como un diccionario, donde las claves son los campos del dataset ("nombre", "poblacion", "superficie", "continente") y los valores son los datos específicos de ese país. Esto permite una forma semántica y eficiente de acceder a los atributos de un país, por ejemplo, pais["poblacion"].

2. Estructuras de Control y Modularización

- **Funciones** Las Funciones son bloques de código reutilizable, definidos con la palabra reservada def, que ejecutan una tarea específica. Facilitan la modularización del código, lo que mejora la legibilidad, el mantenimiento y permite aplicar el principio de "una función = una responsabilidad".
 - **Aplicación en el proyecto:** El proyecto está completamente modularizado en diferentes archivos (.py), donde cada archivo agrupa funciones por responsabilidad:
 - crud.py: Funciones para Crear/Cargar/Actualizar los datos en el CSV (cargar_datos(), actualizar_csv()).
 - ordenamiento.py: Funciones para ordenar (ord_nombre(), ord_superficie()).
 - validaciones.py: Funciones para garantizar la integridad de los datos de entrada (num_no_negativo(), texto_vacio()).

- **Condicionales** Las Sentencias Condicionales (if, elif, else) permiten controlar el flujo de ejecución del programa al decidir qué bloque de código ejecutar en función de si una o más expresiones son Verdaderas (True) o Falsas (False). Python 3.10+ también incorpora la estructura match/case para un manejo de múltiples condiciones de manera más legible, similar a un menú.
 - **Aplicación en el proyecto:**
 - **Validaciones:** En validaciones.py, se usan bucles while e if/else para forzar al usuario a ingresar datos válidos (ej. un número no negativo).
 - **Filtrado:** En filtrar_paises.py, se usan if con operadores lógicos (and) para verificar si un país cumple con múltiples criterios de filtro, como estar dentro de un rango de población: `if p_menor <= pais["poblacion"] and p_mayor >= pais["poblacion"]`.
 - **Menús:** En principal.py y estadistica.py, la estructura match/case o if/elif/else se utiliza para seleccionar la función a ejecutar según la opción elegida por el usuario.

3. Técnicas de Procesamiento de Datos

- **Ordenamientos** El Ordenamiento es el proceso de disponer los elementos de una colección según un criterio predefinido (ascendente o descendente). Python facilita esta tarea con la función incorporada `sorted()`, que retorna una nueva lista ordenada a partir de cualquier iterable.
 - **Aplicación en el proyecto:** Las funciones en ordenamiento.py utilizan la función `sorted()`. Para ordenar la lista de diccionarios, se extraen primero solo los valores de la clave deseada (ej., "nombre" o "superficie") en una lista temporal, se ordena esta lista, y luego se recorre la lista original para imprimir los países en el orden correcto.
- **Estadísticas Básicas** Las estadísticas básicas son cálculos matemáticos aplicados a un conjunto de datos. En el contexto del proyecto, esto incluye medidas de tendencia central y posición extrema.
 - **Aplicación en el proyecto:**
 - **Promedio (Media Aritmética):** El proyecto calcula el promedio (suma de todos los valores dividida por el número de elementos) de la población y la superficie de todos los países, implementado en la función `promedio()` de estadistica.py.
 - **Máximo y Mínimo:** Se busca el país con la mayor y la menor población a través de la función `pais_mayor_menor_poblacion()` en estadistica.py, utilizando variables auxiliares para rastrear el valor extremo y el nombre del país asociado durante la iteración.
- **Archivos CSV** CSV (Comma-Separated Values - Valores Separados por Comas) es un formato de archivo de texto simple y tabular. Se utiliza comúnmente para el intercambio de datos entre programas (ej. bases de datos y hojas de cálculo). Los datos están separados por comas (o a veces punto y coma).

- **Aplicación en el proyecto:** El proyecto utiliza el módulo estándar de Python llamado csv para interactuar con el archivo paises.csv.
 - Las funciones en crud.py emplean csv.DictReader para leer el archivo CSV y cargarlo directamente como una lista de diccionarios en memoria (usando los encabezados como claves).
 - Para escribir o actualizar, se usa csv.DictWriter, lo que garantiza que los datos se guarden en el formato correcto (nombre,poblacion,superficie,continente).
-

Bibliografía

. Python Software Foundation. (2025). *The Python 3 Tutorial*. Recuperado el 15 de noviembre de 2025, de <https://docs.python.org/3/tutorial/>

. Python Software Foundation. (2025). *csv — CSV File Reading and Writing*. Recuperado el 15 de noviembre de 2025, de <https://docs.python.org/3/library/csv.html>

. Python Software Foundation. (2025). *statistics — Mathematical statistics functions*. Recuperado el 15 de noviembre de 2025, de <https://docs.python.org/3/library/statistics.html>

. Lutz, M. (2013). *Learning Python* (5ª ed.). O'Reilly Media.