

Trabajo Práctico Integrador

Alumnos

Jeremías Luzuriaga

Tecnicatura Universitaria en Programación - Universidad Tecnológica Nacional.

Docente Titular

Cinthia Rigoni y Ramiro Hualpa

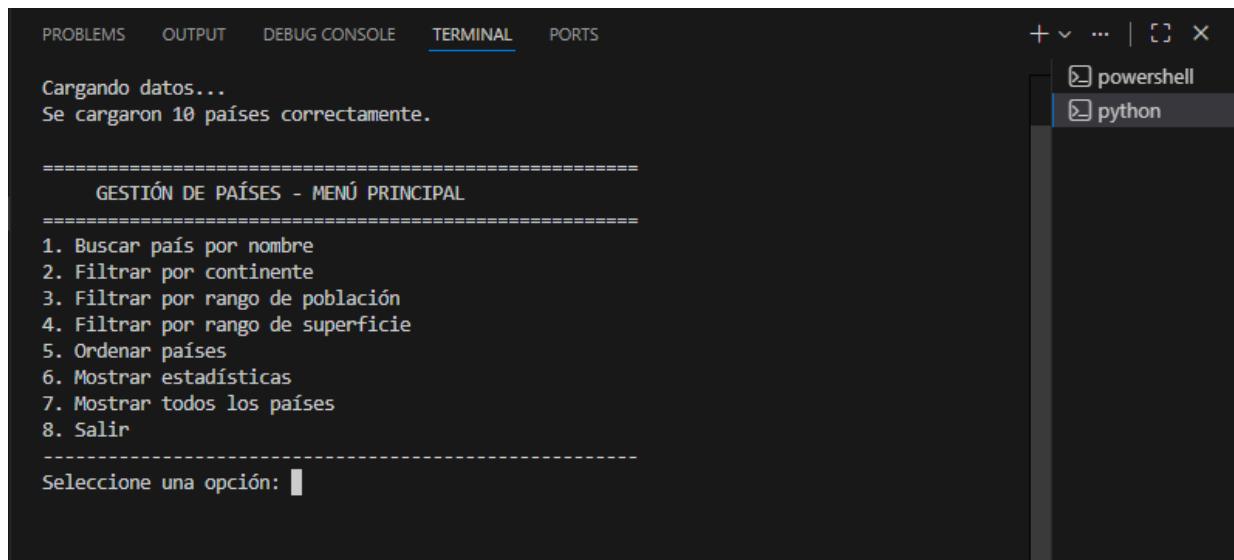
Tabla de contenido

Introducción -----	3
Objetivos -----	4
Desarrollo -----	5
Listas -----	5
Diccionarios -----	6
Funciones -----	7
Condicionales -----	7
Ordenamientos -----	8
Estadísticas base -----	10
Archivos CSV -----	11
Diagrama de flujo -----	12
Conclusión -----	13
Referencias -----	14

Trabajo Práctico N°01

Introducción

Este trabajo tiene como finalidad mostrar cómo se gestiona la información de países utilizando Python utilizando lo aprendido en el semestre: condicionales, funciones, listas, bucles, diccionarios, archivos CSV y cálculos básicos.



The screenshot shows a terminal window with the following content:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
+  ... | powershell
powershell
python

Cargando datos...
Se cargaron 10 países correctamente.

=====
GESTIÓN DE PAÍSES - MENÚ PRINCIPAL
=====
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. Mostrar todos los países
8. Salir

Seleccione una opción: █
```

The terminal window has tabs at the top: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS. On the right side, there is a sidebar with icons for powershell and python, with python currently selected. The main area displays a message about loading data, a menu of 8 options for managing countries, and a prompt for selecting an option.

OBJETIVOS

Desarrollar un sistema en Python que permita administrar, consultar y analizar información de distintos países de manera eficiente. A través de una interfaz por consola, el usuario podrá realizar búsquedas, aplicar filtros, ordenar datos y obtener estadísticas relevantes a partir de un conjunto de datos almacenado en un archivo CSV.

Este proyecto busca aplicar los conocimientos adquiridos en la materia de Programación 1, integrando estructuras de datos como listas y diccionarios, funciones, condicionales, bucles, manejo de archivos y modularización del código.

Desarrollo

CONCEPTOS

Listas

Una lista es una estructura de datos mutable, ordenada y heterogénea que permite almacenar múltiples elementos en una sola variable.

En este trabajo se utiliza para almacenar todos los países leídos del CSV

```
== Todos los países (10 países) ==
Nombre: Argentina
Población: 45,376,763 habitantes
Superficie: 2,780,400 km2
Continente: América
-----
Nombre: Japón
Población: 125,800,000 habitantes
Superficie: 377,975 km2
Continente: Asia
-----
Nombre: Brasil
Población: 213,993,437 habitantes
Superficie: 8,515,767 km2
Continente: América
-----
Nombre: Alemania
Población: 83,149,300 habitantes
Superficie: 357,022 km2
Continente: Europa
-----
Nombre: China
Población: 1,444,216,107 habitantes
Superficie: 9,596,961 km2
Continente: Asia
```

Las ventajas que tiene son:

- Mantiene el orden de carga
- Permite el acceso por índice (aunque no se usa acá)
- Ideal para iterar y filtrar

Diccionarios

Un diccionario es una colección de pares clave-valor, donde las claves son únicas e inmutables. Cada clave (key) tiene asociado un valor (value), y se accede a los datos usando la clave, no el índice como en las listas. Estas son flexibles, pueden guardar distintos tipos de datos de manera clara.

En el proyecto, cada país es un diccionario, en donde para acceder rápido se usa una clave. Esto lo convierte en una estructura ideal para CSV con encabezados. Tiene legibilidad, haciendo que sea más claro poner “pais[“nombre”]” en lugar de “pais[0]”. Tiene flexibilidad ya que permite quitar o agregar campos con facilidad y es compatible con csv.DictReader

```
=====
GESTIÓN DE PAÍSES - MENÚ PRINCIPAL
=====
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. Mostrar todos los países
8. Salir

-----
Seleccione una opción: 1
Nombre del país (parcial): chi

*** Búsqueda: 'chi' (2 países) ***
Nombre: China
Población: 1,444,216,107 habitantes
Superficie: 9,596,961 km2
Continente: Asia

-----
Nombre: Chile
Población: 19,116,201 habitantes
Superficie: 756,102 km2
Continente: América
-----
```

Funciones

Las funciones son bloques de código reutilizables con parámetros y valor de retorno la cuales se crean usando la palabra clave “def”. Cada función tiene un nombre que indica qué hace, tiene parámetros los cuales sirven para pasarle datos con el fin de que trabaje con ellos y pueden devolver un resultado con “return”.

Sirven para organizar el código en bloques claros y legibles, evitan la duplicación del código, facilitan el mantenimiento y la modularización y permiten dividir un problema grande en subproblemas más pequeños.

En el proyecto se ha usado la modularización en 6 archivos y 10+ funciones en donde cada función tiene una responsabilidad:

- cargar_paises() = solo lee CSV
- mostrar_lista() = solo imprime
- ordenar() = solo ordena

Condicionales

Permiten que el programa tome decisiones en donde ejecutan cierto bloque de código solo si se cumple una condición lógica el tipo verdadero/falso.

En Python se usan principalmente con las palabras clave:

- If = si la condición es verdadera, ejecuta el bloque
- Elif = si no se cumplió lo anterior, prueba otra condición
- Else = si ninguna condición se cumplió, ejecuta este bloque

Estos permiten controlar el flujo del programa, evitan errores de validación y hacen que el programa sea interactivo y dinámico.

En el proyecto se han usado en la validación de datos como por ejemplo en el menú “if opcion == “1”: buscar_pais().

```
===== GESTIÓN DE PAÍSES - MENÚ PRINCIPAL =====
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. Mostrar todos los países
8. Salir

Seleccione una opción: 3
Población mínima: 4000000
Población máxima: 46000000

== Población: 40,000,000 - 46,000,000 (1 países) ==
Nombre: Argentina
Población: 45,376,763 habitantes
Superficie: 2,788,400 km2
Continente: América
===== GESTIÓN DE PAÍSES - MENÚ PRINCIPAL =====
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. Mostrar todos los países
8. Salir

Seleccione una opción: 3
Población mínima: 5000000
Población máxima: 10000000
No se encontraron resultados.
```

Ordenamientos

Es el proceso de organizar los elementos en una lista según un criterio:

- Alfabético
- Numérico

- Otro criterio definido por el programador

En Python se usan principalmente las funciones:

- `Sorted()` = devuelve una nueva lista ordenada
- `.sort()` = ordena la lista en el lugar (modifica la original)

También se pueden ordenar listas de diccionarios indicando qué campo usar como criterio. En este proyecto, un ejemplo claro es cuando se ordena a los países por población.

```
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. Mostrar todos los países
8. Salir
-----
Seleccione una opción: 5

Ordenar por:
1. Nombre (A-Z)
2. Nombre (Z-A)
3. Población (mayor a menor)
4. Población (menor a mayor)
5. Superficie (mayor a menor)
6. Superficie (menor a mayor)
Opción: 3

== Ordenado por población desc (10 países) ==
Nombre: China
Población: 1,444,216,107 habitantes
Superficie: 9,596,961 km2
Continente: Asia
-----
Nombre: India
Población: 1,380,004,385 habitantes
Superficie: 3,287,590 km2
Continente: Asia
-----
Nombre: Brasil
Población: 213,993,437 habitantes
Superficie: 8,515,767 km2
Continente: América
```

Estadísticas básicas

Son cálculos simples que permiten resumir y entender mejor los datos. En Python, se aplican sobre listas de números en donde las más comunes son:

- Máximo (max) = el valor más grande
- Mínimo (min) = el valor más pequeño
- Promedio (sum/len) = la media aritmética
- Conteo (len) = cuántos elementos hay

En el proyecto, cada país es un diccionario con nombre, población, superficie, continente. Esto hace que sea posible calcular estadísticas como por ejemplo cuál país es más grande.

```
=====
GESTIÓN DE PAÍSES - MENÚ PRINCIPAL
=====
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. Mostrar todos los países
8. Salir
-----
Seleccione una opción: 6
-----
ESTADÍSTICAS
-----
País con mayor población: China (1,444,216,107)
País con menor población: Chile (19,116,201)
Promedio de población: 355,793,906 habitantes
Promedio de superficie: 4,460,697 km2
Países por continente:
    • América: 4
    • Asia: 3
    • Europa: 3
-----
```

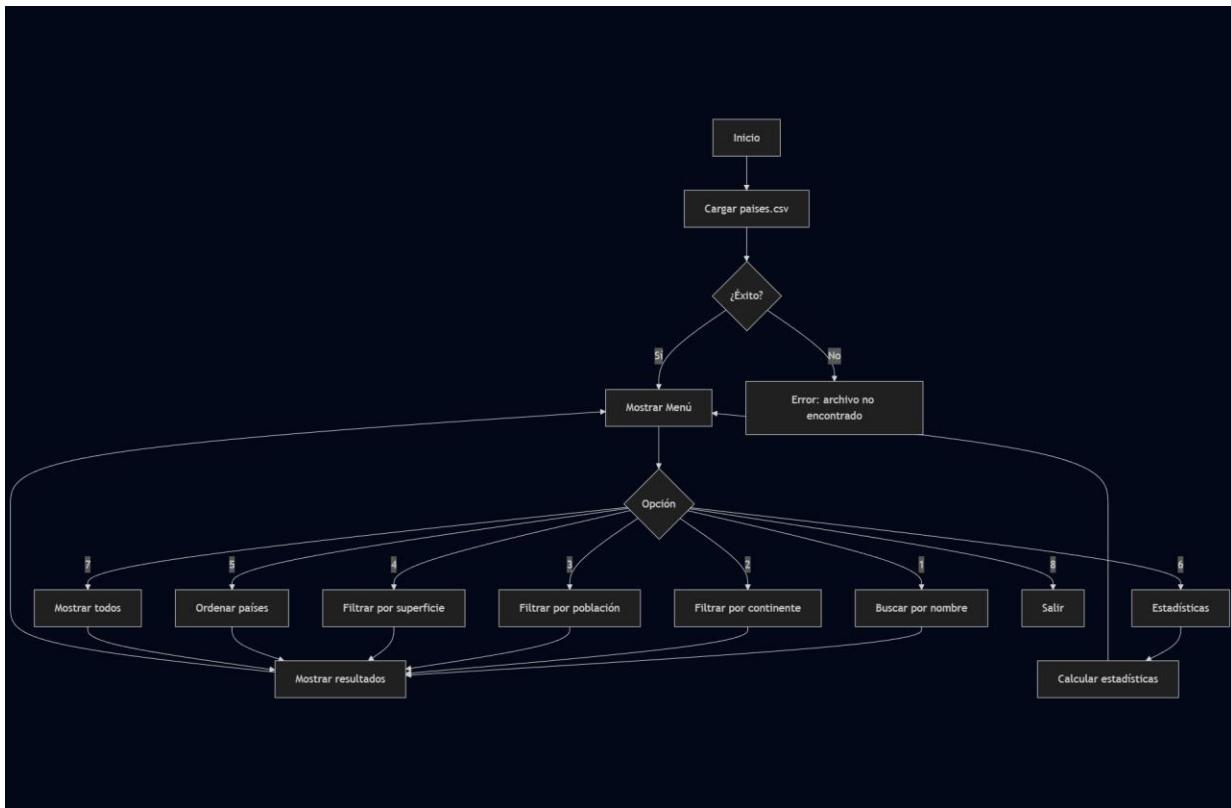
Archivos CSV

Es un formato de texto plano donde los datos se organizan en filas y columnas, separadas por comas. Tiene uso continuo para datasets debido a que es simple, liviano y compatible con casi todos los programas.

En el proyecto, cada fila es un país, y cada columna es un dato: nombre, población, superficie y continente

Para leerlo, Python tiene el módulo CSV que lo hace muy fácil en donde se escribe “import csv” y de ahí se llama al texto (en este caso “paises.csv”)

	países.csv
1	nombre,poblacion,superficie,continente
2	Argentina,45376763,2780400,América
3	Japón,125800000,377975,Asia
4	Brasil,213993437,8515767,América
5	Alemania,83149300,357022,Europa
6	China,1444216107,9596961,Asia
7	Francia,67399000,551695,Europa
8	Chile,19116201,756102,América
9	Rusia,145912025,17098242,Europa
10	India,1380004385,3287590,Asia
11	Perú,32971846,1285216,América

Diagrama de flujo

Link del diagrama hecho en la página de mermaid

[Online FlowChart & Diagrams Editor - Mermaid Live Editor](#)

Conclusión

Este trabajo ha sido una experiencia muy enriquecedora para mí. A través del desarrollo de la aplicación en Python, consolidé los conceptos básicos de la materia Programación 1 de una manera práctica y real.

Aprendí a cómo usar listas y diccionarios para organizar datos de forma clara y eficiente, comprendí la importancia de dividir el código en funciones pequeñas logrando que sea todo más fácil de entender y corregir, a leer archivos CSV y trabajar con datos reales, a ordenar y calcular estadísticas con funciones simples de Python sin necesidad de librerías externas, entre otros.

En lo personal me ayudó a ganar confianza al ver que puedo crear un programa completo desde cero: desde leer un archivo hasta mostrar resultados bien estructurados en consola. Aprendí a organizar mejor mi código, algo que antes me costaba y dificultaba comprender, y ahora veo lo útil que es para no perderme. Comprendí no solamente que es programar, sino también a entender cómo debe pensar un programador y como esto me puede ayudar en mi carrera profesional.

Referencias

[Operadores_logicos.ipynb - Colab](#)

[dicionales%20-%20Programaci%C3%B3n%20I.pdf?forcedownload=1](https://campus.frm.utn.edu.ar/pluginfile.php/428891/mod_resource/content/1/PDF%20Con)