Trabajo Práctico Integrador – Programación I

Tema: Gestión de Datos de Países en Python
Carrera: Tecnicatura Universitaria en Programación – UTN
Modalidad: A Distancia
Año: 2025
Alumno: Lautaro Ojeda Franco Tarditi

Marco Teórico

El desarrollo del presente trabajo se basa en la aplicación de los conceptos fundamentales de la programación estructurada, con énfasis en el lenguaje **Python 3**. El proyecto busca integrar de manera práctica los contenidos abordados a lo largo de la cursada, tales como estructuras de datos, modularización, persistencia de información y generación de estadísticas.

En Python, una **lista** es una colección ordenada y mutable que permite almacenar múltiples elementos, incluso de distintos tipos. En este proyecto se utiliza para mantener en memoria los registros de países cargados desde un archivo CSV. Cada elemento representa un país y puede manipularse mediante iteraciones o filtrados.

Los **diccionarios** son estructuras de datos que almacenan pares de clave-valor. En este trabajo, cada país se representa mediante un diccionario con claves semánticas (`nombre`, `población`, `superficie`, `continente`), lo que facilita la lectura y el acceso a los datos.

La **modularización del código** se logra a través del uso de funciones, cada una con una responsabilidad específica, como la carga de datos, el filtrado, el ordenamiento o el cálculo de estadísticas. Esta organización se alinea con el principio de “una función, una tarea”, mejorando la legibilidad y la reutilización del código.

El uso de **estructuras condicionales** (`if`, `elif`, `else`) permite la toma de decisiones en función de las entradas del usuario y de las condiciones lógicas del programa. Las **estructuras repetitivas** (`for`, `while`) facilitan la iteración sobre los datos o la permanencia en el menú interactivo.

La **persistencia de datos** se implementa mediante la lectura y escritura de archivos CSV. Se utiliza la instrucción `with open(...)` para garantizar el cierre seguro del archivo y prevenir errores de acceso. La lectura transforma cada línea en una estructura de datos manipulable dentro del programa.

Finalmente, el proyecto incorpora la librería **Rich**, que permite presentar los resultados en formato tabular y con estilos visuales claros,como a su vez se agregó la **API** donde se alojan los países **(REST countries)**.

Fuentes consultadas:

- Python Software Foundation. (2024). *Python 3 Documentation*. https://docs.python.org/3/

- Textualize. (2024). *Rich Library Documentation*. https://rich.readthedocs.io/

- Universidad Tecnológica Nacional. (2025). *Apuntes de Programación I*.

- API - REST countries. https://restcountries.com/

Objetivo del Trabajo

El objetivo general es desarrollar una aplicación en Python que permita **gestionar información sobre países**, aplicando estructuras de datos dinámicas, técnicas de modularización y principios de programación estructurada.

Entre los objetivos específicos se destacan:

- Implementar funciones que permitan **buscar, filtrar, ordenar y analizar** datos en un archivo CSV.

- Aplicar **listas y diccionarios** para representar y manipular los registros en memoria.

- Desarrollar una **interfaz de consola interactiva** utilizando la librería *Rich*.

- Incorporar **validaciones** para garantizar la estabilidad del sistema y prevenir errores de entrada.

- Generar **estadísticas básicas** a partir de los datos cargados.

Diseño del Caso Práctico

El proyecto se denomina *Gestión de Datos de Países* y consiste en un sistema que opera sobre un archivo `paises.csv`. Cada registro contiene el nombre del país, su población, superficie y continente.

La aplicación presenta un menú principal que permite al usuario realizar las siguientes operaciones:

- Buscar un país por nombre (coincidencia parcial o exacta).
- Filtrar países por continente, rango de población o rango de superficie.
- Ordenar países por nombre, población o superficie en ambos sentidos (ascendente/descendente).
- Calcular estadísticas globales (máximo, mínimo, promedio y distribución por continente).
- Reiniciar el archivo CSV con confirmación.

El flujo del programa se estructura en dos módulos principales:

1. **main.py:** controla el menú, las opciones y la interacción con el usuario.
2. **funciones.py:** contiene las funciones de lectura, escritura, filtrado, ordenamiento y estadísticas.

Esta división favorece la **modularidad** y facilita el mantenimiento y la ampliación futura del sistema.

Metodología Utilizada

El desarrollo siguió una metodología incremental, comenzando con la definición del dataset base (`paises.csv`) y la creación de funciones independientes para cada operación.
Se realizaron pruebas unitarias básicas en consola para verificar el comportamiento de las funciones antes de integrarlas en el menú principal.

La comunicación entre los módulos se maneja mediante la importación de funciones, lo que permite mantener el código limpio y reutilizable.
Cada bloque fue documentado mediante comentarios breves, describiendo su propósito y parámetros.

El formato de salida fue diseñado con *Rich*, aprovechando paneles y tablas para presentar resultados claros y profesionales.

Resultados Obtenidos

El sistema cumple con todos los requerimientos planteados en las consignas del TPI:

- Permite agregar, filtrar, buscar y ordenar países correctamente.
- Presenta estadísticas precisas, calculadas a partir de los datos del archivo CSV.
- Valida entradas y evita errores comunes durante la ejecución.
- Garantiza la persistencia de la información y el cierre correcto de archivos.

Ejemplo de salida en consola:

```
Seleccione una opción: 3

📊 ESTADÍSTICAS

📊 ESTADÍSTICAS GLOBALES

Total de países          3
País más poblado         Federative Republic of Brazil (212,559,409)
País menos poblado       Argentine Republic (45,376,763)
Población promedio       108,383,822 hab
Superficie promedio      38,463,557 km²

🌍 DISTRIBUCIÓN POR CONTINENTE

Continente    Cantidad    Porcentaje
South America 2 países    66.7%
Europe        1 países    33.3%
```

El ejemplo deja ver la implementación de librerías que le da una estética visual y se puede ver que contiene información de filtrado la cuál es comprobable en fuentes oficiales, como por ejemplo: Tasa de población, Superficie, Continente.

Resultados y Evaluaciones

A la hora de hacer el proyecto pusimos en pie las funciones las cuales funcionario bien, se hizo un menu dentro de funciones para comprobar que estuvieran bien, tuvimos errores con el csv, a la hora de cargar archivos no se guardaban correctamente o directamente nunca se creaba el archivo dentro del csv. La comunicación del equipo se dio en entorno virtual y

se pudieron concretar las bases del código gracias a la division del código y la comunicacion.

Conclusiones

El desarrollo del proyecto permitió aplicar de manera integral los contenidos de la materia Programación I, por cuenta propia pudimos aprender a manejarnos libremente con librerías y APIs aplicándola al código sin problemas, a su vez pudimos aprender a crear un entorno virtual donde se pueda recorrer un lenguaje de programación sin necesidad de tener que descargar todas sus versiones, librerías, apis, etc.

La experiencia de implementación mostró la importancia de diseñar el código de forma ordenada y escalable, favoreciendo la lectura y la reutilización. Nos parece importante tanto librerías, apis y docker porque se ahorra tiempo, se evitan errores, facilita el mantenimiento del código, interoperabilidad, entornos consistentes y colaboración fluida.

La división del trabajo se realizó con el objetivo de optimizar el desarrollo del proyecto, respetando los principios de modularidad, claridad y colaboración. Se dividió en 2 partes, menú y funciones. Esto permite realizar por parte del menú: la interfaz, lógica de navegación y validación de entrada. Por parte de las funciones: permite la implementación de la lógica, diseño modular y funciones reutilizables.