

T.P. Integrador – Programación 1

Gestión de Datos de Países en Python

Tecnicatura Universitaria en Programación

Integrantes:

Adrián González, Comisión 6

Ezequiel Taboada, Comisión 13

octubre 2025

Introducción

El presente trabajo práctico integrador tiene como objetivo aplicar los conceptos fundamentales de la materia Programación 1 mediante el desarrollo de una aplicación en Python que permite gestionar información sobre países. El sistema implementa operaciones de búsqueda, filtrado, ordenamiento y generación de estadísticas a partir de un archivo CSV, integrando el uso de listas, diccionarios, funciones, estructuras de control y modularización del código.

En esta nueva versión se incorporó una interfaz gráfica con Tkinter para la carga de archivos, lo cual mejora la experiencia del usuario. También se fortaleció el control de errores y se amplió la organización modular del sistema.

Objetivos

- Implementar un sistema de gestión de países con Python.
- Reforzar el uso de estructuras de datos como listas y diccionarios.
- Practicar la modularización del código dividiendo el sistema en distintos archivos.
- Aplicar manejo de errores y validaciones en la carga y procesamiento de datos.
- Incorporar una interfaz gráfica para la selección de archivos.
- Desarrollar estadísticas y operaciones útiles sobre la información.

Diseño del programa

El sistema se diseñó con una estructura modular, dividiéndose en distintos archivos que cumplen responsabilidades específicas:

- a_main.py: control principal del flujo y menús.
- b_funciones_csv.py: funciones de carga de archivos CSV.
- c_funciones_colecciones.py: operaciones sobre colecciones de países (filtros, ordenamientos, etc.).
- d_modelo.py: definición del modelo de datos de un país.
- e_funciones_generales.py: utilidades generales para interacción con el usuario y visualización.
- paises.csv: dataset inicial con la información base.

Este diseño modular mejora la organización, legibilidad y mantenibilidad del código.

Descripción del sistema

El sistema permite al usuario gestionar información de países contenida en un archivo CSV. Ofrece las siguientes funcionalidades:

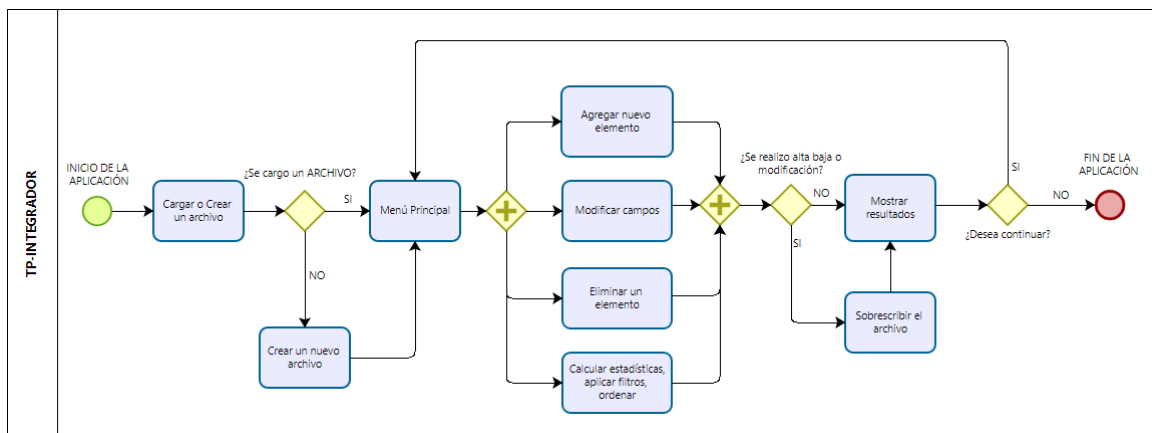
1. Cargar archivo CSV mediante un cuadro de diálogo gráfico o Crear uno nuevo.
2. Agregar un país.

3. Editar población y/o superficie.
4. Buscar países por coincidencia parcial o exacta en el nombre.
5. Filtrar países por continente, rango de población o superficie.
6. Ordenar países por nombre, población o superficie, en orden ascendente o descendente.
7. Mostrar estadísticas: mayor y menor población, mayor y menor superficie, promedios de población y superficie por continente, y cantidad de países por continente.
8. Eliminar un país.
9. Salir del programa.

Flujo de operaciones

El flujo de operaciones del sistema sigue los siguientes pasos:

- Inicio del programa.
- Selección del archivo CSV mediante interfaz gráfica o creación de uno nuevo.
- Presentación del menú principal con opciones de creación, actualización búsqueda, filtrado, ordenamiento, estadísticas, eliminación y salida.
- Ejecución de la opción seleccionada.
- Actualización del archivo(solo si hubo modificaciones).
- Retorno al menú principal hasta que el usuario seleccione 'Salir'.
- Fin del programa.



Marco teórico

❖ Listas:

“Las listas permiten almacenar objetos mediante un orden definido y con posibilidad de duplicados. Las listas son estructuras de datos mutables, lo que significa que podemos añadir, eliminar o modificar sus elementos.”

Fuente: [Aprende Python](#)

En la app usamos en varias ocasiones listas, por ejemplo, para generar las opciones del menú y almacenar la colección de países.

❖ Diccionarios:

“Un Diccionario es una colección desordenada y variable de valores de datos que contiene pares clave-valor. Cada par clave-valor en el diccionario asigna la clave a su valor asociado, haciéndolo más optimizado. Un diccionario en Python se declara encerrando una lista separada por comas de pares clave-valor usando llaves ({}). El Diccionario Python se clasifica en dos elementos: claves y valores.

- *Las claves serán un solo elemento único e irrepetible.*
- *Los valores pueden ser una lista o una lista dentro de una lista, números, etc.”*

Fuente: [Python-3.com](#)

En la app lo usamos para representar a cada país, usando las claves para el nombre de cada campo y los valores para el contenido de cada campo.

❖ Funciones:

“Las funciones son bloques de código reutilizables que realizan una tarea específica. Son uno de los conceptos fundamentales de la programación y te permiten organizar tu código de manera más clara, evitar repetición y facilitar el mantenimiento.”

Fuente: [El Pythonista](#)

En la app el uso de funciones fue fundamental, ya que se encuentran presentes en toda la app, desde mostrar resultados, validar las entradas de datos, buscar, editar, calcular estadísticas hasta trabajar con archivos .CSV.

❖ Condicionales:

“Los condicionales son estructuras de control que permiten que un programa tome decisiones basadas en ciertas condiciones.

Estas sentencias nos permiten evaluar expresiones y ejecutar diferentes bloques de código dependiendo de si estas expresiones son verdaderas o falsas.

En Python, los condicionales se implementan principalmente a través de las sentencias if, elif (abreviatura de “else if”), y else.”

Fuente: [Luis Llamas](#)

En la app usamos condicionales en varios puntos, por ejemplo, para verificar si hay un archivo cargado, si el nombre de un país coincide con el buscado, si los datos ingresados son los esperados, etc.

- ❖ **Ordenamientos:** *“Los algoritmos de ordenación son fundamentales en la programación, ya que permiten organizar los elementos de una lista o arreglo en un orden específico, como ascendente o descendente. Algunos de los algoritmos de ordenación más comunes incluyen el algoritmo de burbuja, la ordenación por inserción, la ordenación por selección, el algoritmo quicksort y el mergesort.”*

Fuente: [Juan Antonio Ripoll](#)

“Las listas de Python tienen un método incorporado list.sort() que modifica la lista in situ. También hay una función incorporada sorted() que crea una nueva lista ordenada a partir de un iterable.”

Fuente: [Python](#)

En la app optamos por usar el método sorted() que viene incorporado en las listas, ya que de una forma simple podemos pasar el nombre del campo y el criterio, esto nos retorna una copia de la lista original.

- ❖ **Estadísticas básicas:**

“Las funciones min() y max() integradas en Python resultan útiles cuando necesitas encontrar las funciones más pequeñas y grandes. Aunque puedan parecer cálculos bastante básicos, resultan tener muchos casos de uso interesantes en la programación del mundo real.”

Fuente: [Python-3](#)

*“La función **mean()** del módulo de Python se utiliza para calcular el promedio de un conjunto de valores numéricos. Suma todos los valores de una lista y divide el total por el número de elementos.”*

Fuente: [Geeks For Geeks](#)

En la app usamos las funciones:

- `min()` □ para calcular el país con menor población y superficie.
- `max()` □ para calcular el país con mayor población y superficie.
- `mean()` □ para calcular el promedio de población y superficie.

❖ Archivos CSV:

“CSV significa valores separados por comas. Un archivo CSV se puede abrir en Google Sheets o Excel y se formateará como una hoja de cálculo. Sin embargo, un archivo CSV es en realidad un archivo de texto plano. También se puede abrir con un editor de texto como Atom.

Los archivos CSV nos ofrecen una forma sencilla y eficaz de organizar datos sin necesidad de usar un programa de base de datos. Leer y escribir archivos CSV con Python es muy fácil.”

Fuente: [Python para principiantes](#)

En la app utilizamos archivos .CSV para asegurar la persistencia de los datos,

❖ Tkinter:

“Tkinter es una librería del lenguaje de programación Python y funciona para la creación y el desarrollo de aplicaciones de escritorio. Esta librería facilita el posicionamiento y desarrollo de una interfaz gráfica de escritorio con Python. Tkinter es el paquete estándar de Python para interactuar con Tk.”

Fuente: [Keep Coding](#)

“Tk provee funciones para desplegar diálogos para examinar archivos o carpetas en Windows, Linux (o cualquier otro sistema basado en Unix) y macOS. En Python las funciones son accesibles a través del módulo `tkinter.filedialog`”

Fuente: [Recursos Python](#)

En la app usamos Tkinter para generar una interfaz grafica al momento de cargar el archivo .CSV, esto facilita la selección del archivo y evita tener que escribir la ruta a mano.

Conclusiones

Este trabajo nos permitió descubrir y aplicar herramientas fundamentales del lenguaje Python. Aprendimos a utilizar listas y diccionarios para organizar datos, funciones para dividir el código en partes reutilizables, y estructuras condicionales para tomar decisiones dentro del programa. También exploramos cómo realizar ordenamientos, aplicar filtros y generar estadísticas básicas, lo que nos ayudó a entender cómo se puede analizar información de manera automatizada.

Estas herramientas nos parecen importantes porque permiten resolver problemas reales de forma eficiente. Por ejemplo, poder filtrar países por continente o calcular el promedio de población nos mostró cómo la programación puede facilitar tareas que serían muy tediosas de hacer manualmente.

Justificamos el uso de estas herramientas porque cada una cumple un rol específico en el sistema: las listas almacenan los países, los diccionarios representan sus atributos, las funciones organizan el código, y las estructuras condicionales permiten controlar el flujo del programa. Un ejemplo claro fue el uso de `sorted()` con claves personalizadas para ordenar países por superficie o población, lo que nos permitió obtener resultados precisos con pocas líneas de código.

La organización del trabajo entre nosotros fue clave. Dividimos las tareas de forma equitativa: uno se encargó del diseño del sistema y la estructura del código, mientras que el otro se ocupó de la carga de datos, pruebas y documentación. Esta división nos permitió avanzar de manera ordenada y aprender colaborativamente, compartiendo dudas y soluciones a lo largo del desarrollo.

En resumen, este proyecto nos ayudó a comprender cómo aplicar la programación a un caso práctico, y nos motivó a seguir explorando.