

Trabajo Práctico Integrador

Informe Teórico

Materia: Programación I

Carrera: Tecnicatura Universitaria en Programación

Año: 2025

Integrantes: Franco Storani, Facundo Degen

Introducción

El presente informe tiene como objetivo la exposición de los pasos y recursos utilizados en la realización del trabajo práctico integrador de la materia Programación 1 explicando cómo se usaron las librerías de Python y manejo de datos relacionados con el manejo de listas, diccionarios, funciones, condicionales, estructuras repetitivas, ordenamientos, estadísticas y archivos CSV.

Se buscó implementar los conceptos dentro de un entorno web utilizando Flask y una interfaz html, logrando una presentación visual de los resultados con una interacción entre el backend y frontend.

Desarrollo teórico de los conceptos aplicados

Listas

Las listas son estructuras de datos que permiten almacenar múltiples elementos en una sola variable, manteniendo un orden determinado.

En el proyecto se utilizaron para contener la información de los países leídos desde el archivo CSV, donde cada país fue representado como un diccionario dentro de una lista general.

Gracias a esta estructura, fue posible recorrer la colección mediante bucles for para aplicar filtros, búsquedas y cálculos estadísticos. Por ejemplo, se recorrió la lista completa para identificar países que pertenecían a determinado continente o que cumplieran con rangos de población específicos.

Diccionarios

Los diccionarios fueron empleados para representar los registros individuales de cada país, asociando y valores de acuerdo a su descripción (por ejemplo: nombre, población, superficie, continente).

Esta estructura permitió un acceso eficiente a los datos y una manipulación clara, ya que se puede acceder a cada país desde su clave dentro del diccionario

Además, su combinación con listas permitió construir estructuras jerárquicas donde cada elemento representaba un país con todos sus atributos, lo cual facilitó la búsqueda, filtrado y ordenamiento.

Funciones

Se dividió al programa en módulos a través del uso de funciones, asignando a cada una un proceso diferente.

Se crearon funciones para leer el archivo CSV, realizar filtros por continente, buscar coincidencias por nombre, calcular estadísticas y ordenar los registros.

El uso de funciones mejoró la legibilidad, la reutilización del código y el mantenimiento general del sistema. Cada módulo cumplió una tarea concreta, lo que facilitó tanto la depuración como la ampliación de funcionalidades.

Condicionales

Las estructuras condicionales permitieron ser utilizadas como interruptores estableciendo criterios para decidir qué acciones ejecutar ante determinadas condiciones.

Por ejemplo, se utilizaron para verificar si un país pertenecía a un continente determinado o si una población se encontraba dentro de un rango específico.

Asimismo, las validaciones de entrada y los mensajes de error se basaron en sentencias if-else, asegurando que el sistema respondiera adecuadamente ante entradas inválidas o búsquedas sin resultados.

Ordenamientos

El sistema implementó ordenamientos tanto por nombre, población como por superficie, empleando la función `max()` y `min()` para obtener el país con mayor y menor cantidad de población

Esta técnica permitió ofrecer al usuario diferentes modos de visualizar los datos, ya sea en orden ascendente o descendente, según la opción seleccionada en la interfaz.

El ordenamiento garantizó una mejor presentación de la información, contribuyendo a la claridad y comprensión de los resultados obtenidos por el sistema.

Estadísticas básicas

Las estadísticas se calcularon a partir de las listas de países.

Entre los indicadores obtenidos se incluyeron el país con mayor y menor población, el promedio de población y de superficie, y la cantidad de países por continente.

El objetivo fue obtener una visión general de los datos de manera rápida y precisa, permitiendo comparar características entre regiones.

Archivos CSV

El manejo de archivos CSV fue fundamental para la carga inicial de los datos.

El programa leyó el archivo con el módulo `csv` de Python, convirtiendo cada fila del dataset en un diccionario.

También se creó una función en caso de que no haya un archivo csv creado, se creará.

Se controlaron posibles errores de formato y se garantizaron mensajes claros al usuario en caso de problemas con el archivo.

De esta manera, el sistema se construyó sobre un flujo automatizado de lectura, procesamiento y presentación de datos, manteniendo la trazabilidad y confiabilidad de la información.

Relación con el proyecto Flask y HTML

El proyecto fue desarrollado utilizando el framework Flask, lo que permitió integrar la lógica en Python con una interfaz web diseñada en HTML.

El servidor Flask gestionó las solicitudes de los usuarios, procesó los datos del archivo CSV y devolvió los resultados mediante plantillas HTML.

De esta forma, las funciones implementadas para filtrar, ordenar y calcular estadísticas se invocaron desde las rutas del servidor, mostrando los resultados en páginas estáticas accesibles a través del navegador.

Esta arquitectura favoreció la separación entre la capa lógica y la de presentación, promoviendo buenas prácticas de programación y un flujo de trabajo similar al de las aplicaciones modernas.

El uso de Jinja2 dentro de Flask permitió integrar variables y resultados dentro del HTML, mostrando la información procesada directamente en las vistas del usuario.

Conclusiones

A través del desarrollo de este trabajo integrador, se consolidaron los conocimientos adquiridos en Programación I, especialmente en lo que respecta al manejo de estructuras de datos, modularización del código y procesamiento de información proveniente de archivos externos.

La implementación en Flask permitió comprender cómo las aplicaciones Python pueden interactuar con interfaces web, fortaleciendo la visión integral del desarrollo de software.

El proyecto demostró la importancia de la planificación, la claridad del código y la correcta división de responsabilidades entre funciones, elementos fundamentales para el trabajo colaborativo y el desarrollo profesional futuro.

En suma, se logró un sistema robusto, comprensible y adaptable, que refleja la aplicación práctica de los principios básicos de la programación estructurada y modular en un contexto realista.