TPI Programación 1

Informe Teórico

Gestión de Datos de Países en Python

Alumnos:

Simón Blas – Comisión 2 – simoblas08@gmail.com

Joel Vitrano – Comisión 12 – joevitrano77@gmail.com

DIAGRAMA DEL CODIGO:

https://miro.com/app/board/uXjVJtx2UG0=/

Introducción

Este informe explica los conceptos fundamentales aplicados en el desarrollo del Trabajo Práctico Integrador de “Gestión de Datos de Países en Python”. El proyecto tiene como objetivo reforzar el uso de estructuras de datos, modularización mediante funciones, y técnicas de filtrado, ordenamiento y análisis estadístico sobre un conjunto de datos (dataset) de países almacenado en un archivo CSV.

Conceptos aplicados

Listas

Las listas en Python son estructuras de datos ordenadas y mutables que permiten almacenar múltiples elementos bajo un solo nombre. Son útiles para representar colecciones, como, por ejemplo, un conjunto de países. Cada elemento puede ser accedido mediante su índice y puede modificarse, agregarse o eliminarse.

En Python:

países = [“Argentina”, “Brasil”, “Uruguay”]

Diccionarios

Un diccionario es una estructura de datos que almacena pares clave-valor, lo que permite acceder a los datos de forma más clara y directa. En este proyecto, cada país se representa como un diccionario que incluye su nombre, población, superficie

y continente. Además, nos permiten acceder a fácilmente a los valores mediante sus claves, por ejemplo: pais[nombre] devuelve “Argentina”.

Ejemplo en Python:

pais = {

“nombre” : “Argentina”,

“poblacion” : 45376763,

“superficie” : 2780400,

“continente” : “América”

}

Funciones

Las funciones son bloques de código reutilizables que agrupan una tarea específica. Permiten modularizar el programa, facilitando la lectura, mantenimiento y reutilización del código. Cada función tiene una única responsabilidad, cumpliendo con el principio de claridad. En este proyecto, mientras unas funciones gestionan la lectura y escritura de los datos del archivo CSV, otras realizan búsquedas, filtrados u operaciones de estadística. En el siguiente ejemplo, le enviamos como parámetro una lista a la función, para que trabaje con ella realizando el cálculo del promedio de sus elementos numéricos.

Ejemplo en Python:

def calcular_promedio(lista)

return sum(lista) / len(lista)

Condicionales

Las estructuras condicionales (if, elif, else) permiten tomar decisiones en el flujo del programa. Se utilizan para validar datos, controlar errores y dirigir la ejecución según las condiciones que decidamos establecer. Podemos utilizarlas, por ejemplo, para indicar un aproximado de la población de un país o para validar entradas del usuario.

Ejemplo en Python:

if pais[“poblacion”] > 1000000:

print(“País con más de un millón de habitantes.”)

else:

Print(“País con menos de un millón de habitantes.”)

Ordenamientos

El ordenamiento organiza los elementos de una lista según un criterio, utilizando funciones como sorted() o el método .sort(). Se puede definir la clave del orden, por ejemplo, población o superficie, y el sentido, ascendente o descendente.

Ejemplo en Python:

numeros = [5, 2, 99 11, 7]

numeros_ordenados = sorted(numeros)

Eso ordenara los números de manera ascendente.

Para ordenarlos de forma descendente debemos indicar:

numeros_descendente = sorted(numeros, reverse=True)

Estadísticas básicas

El cálculo de estadísticas básicas permite obtener información resumida de los datos, como país con mayor o menor población, el promedio de la población o superficie y la cantidad de países por continente.

Ejemplo en Python:

promedio_poblacion = sum(pais[“poblacion”] for pais in paises) / len(paises)

Estas operaciones resultan útiles para analizar y comprender nuestro dataset de países de forma cuantitativa.

Archivos CSV

Los archivos CSV o también llamados Comma-Separated Values son archivos de texto que almacenan datos en formato tabular, donde cada línea representa un registro y los valores se separan por comas. El lenguaje de programación Python provee el módulo CSV para leer y escribir este tipo de archivos fácilmente.

Ejemplo en Python:

Import csv

With open(“paises.csv”, “r”, encoding=”utf-8”) as archivo

Lector = csv.DictReader(archivo)

For fila in lector:

print(fila[“nombre”], fila[“poblacion”])

Podemos utilizar DictReader para convertir cada línea del CSV en un diccionario, facilitando su manipulación.

Conclusiones sobre el aprendizaje

La realización de este trabajo integrador nos permitió, como estudiantes, consolidar los fundamentos de la programación estructurada en Python, aplicando de forma práctica los principales conceptos de la cátedra: listas, diccionarios, funciones, condicionales, ordenamientos, estadísticas básicas y manejo de archivos. A través de la implementación de un sistema que gestiona datos reales sobre países, logramos comprender cómo las estructuras de datos se combinan para resolver problemas concretos, permitiendo almacenar, filtrar y analizar información de manera eficiente. Además, la modularización del código mediante funciones favoreció la claridad, la reutilización y el mantenimiento del programa a futuro. El trabajo en equipo, además, también mejoraría la comunicación, la organización y la distribución de tareas durante el desarrollo.

Por otro lado, los ordenamientos y las estadísticas básicas permitieron analizar y extraer información significativa del dataset, favoreciendo la toma de decisiones y la comprensión global de los datos. Finalmente, el manejo de archivos CSV introdujo la noción de persistencia, posibilitando conservar la información más allá de la ejecución del programa.

En conclusión, este proyecto integrador representa un avance significativo en nuestra formación como futuros profesionales en el campo de la programación, ya que combina lógica, buenas prácticas de codificación y análisis de datos, acercándonos a escenarios reales de desarrollo de software y fortaleciendo nuestras competencias técnicas y colaborativas.

Fuentes de información

Material de la cátedra de Programación 1:

- Unidad 3 – Estructuras Condicionales.

https://tup.sied.utn.edu.ar/course/view.php?id=25&section=14

- Unidad 5 - Listas.

https://tup.sied.utn.edu.ar/course/view.php?id=25&section=26

- Unidad 6 – Funciones.

https://tup.sied.utn.edu.ar/course/view.php?id=25&section=34

- Unidad 7 – Datos Complejos.

https://tup.sied.utn.edu.ar/course/view.php?id=25&section=41

- Unidad 8 – Manejo de Archivos.

https://tup.sied.utn.edu.ar/course/view.php?id=25&section=47

Material educativo de Digital House (Aprendizaje de programación imperativa):

- Métodos de ordenamiento:

https://playground.digitalhouse.com/explore/course/programacion-javascript?filters=product_kind%3ADigital+House+One&sortBy=RELEVANCE&search=