

Manipulando grandes cantidades de datos

Proyecto final: Fundamentos de Programación.

Contexto:

Una de las aplicaciones más comunes de un producto de software es la manipulación eficiente de una gran cantidad de datos para producir información. En esta aplicación utilizaremos datos reales sobre aeropuertos, aerolíneas y rutas de todo el mundo, las cuales obtendremos en formato de archivo de texto, para producir archivos de reportes o impresión de información a pantalla.

Objetivo:

El estudiante diseñará e implementará un programa en Python, para generar reportes de datos obtenidos a partir de 3 archivos de texto.

La intención es que, en el diseño e implementación de este programa, se demuestre el conocimiento del uso adecuado de variables, operadores aritméticos, condicionales, ciclos, manipulación de strings, listas, lectura y escritura de archivos de texto.

Descripción:

En la página <https://openflights.org/data.html> se pueden obtener los siguientes archivos de texto: airports.dat (datos de aeropuertos), airlines.dat (datos de aerolíneas) y routes.dat (rutas existentes). Estos tres archivos servirán para llevar a cabo las siguientes acciones en el programa:

1. **Aeropuertos de un país:** Esta sección del programa te arroja a pantalla un listado de todos los aeropuertos de un país con su código y ciudad.
2. **Distancia entre aeropuertos:** Esta sección te pide la ciudad destino y el origen y te arroja la distancia que hay entre los aeropuertos de estas ciudades, si hay más de un aeropuerto, deberá mostrarte el reporte de todos los aeropuertos existentes. La distancia la deberás calcular a través de los datos de longitud y latitud que te proporciona el archivo airports.dat.
3. **Distancia entre aeropuertos de un país:** Se solicita al usuario el nombre de un país, el programa deberá generar un archivo de texto tipo reporte, que informe la distancia que hay entre todos los aeropuertos de ese país. E imprimir a pantalla cuál es el par de aeropuertos que están más alejados entre sí.
Extra: Python tiene muchas y variadas librerías (math o random son librerías) que ayudan o están programadas para hacer cosas muy específicas. Entre ellas cuenta con librerías que te permiten hacer gráficas. Investiga y genera una gráfica tipo histograma para mostrar las distancias entre aeropuertos.
4. **Aerolíneas que van a un destino desde un determinado aeropuerto.** Se solicita al usuario el código del aeropuerto y un destino. El resultado es mostrarnos un reporte de todas las aerolíneas (Nombre y alias) que parten del aeropuerto al destino indicado, indicando si es ruta directa o con escalas. Además de imprimirse a pantalla, se deberá generar un archivo donde en cada línea viene el Nombre y el Alias separado por un guión.

5. **Aerolíneas que operan en un determinado aeropuerto.** Se solicita al usuario el nombre de un aeropuerto y te despliega a pantalla el listado de aerolíneas que tienen actividad (tienen rutas activas) en el aeropuerto indicado.
6. **Destinos de una aerolínea desde un determinado aeropuerto.** En esta sección, el usuario determina la aerolínea y el aeropuerto, la salida es un archivo con información separada por comas que te diga el destino (solo destinos directos) y la distancia hacia ese destino. Además deberá informar a pantalla cuál es el destino más lejano que tiene la aerolínea.

Entradas del programa:

- El usuario debe elegir a partir de un menú la acción desea realizar.
- Archivos .dat obtenidos de la página indicada para la obtención de datos.
- Dependiendo de la opción elegida, solicitar los datos que hagan falta para poder llevar a cabo la funcionalidad.

Salidas:

El programa deberá proporcionar según sea el caso:

- Archivo de texto con el reporte resultante.
- Impresión a pantalla con los datos o reportes solicitados. En cuyo caso, se deben desplegar con un formato ordenado y bien presentado.

NOTA: A todos tus archivos de salida agrégalos como primera línea el título del reporte y la fecha de generación.

Entregables:

1. **Archivo .py por pareja.**
2. Reporte de implementación que incluye:
 - a. Introducción (explicación general del programa).
 - b. Descripción del diseño de funciones, qué funciones tienes y para qué, qué reciben y que devuelven.
 - c. Problemas enfrentados durante la implementación y cómo los resolvieron.
 - d. Conclusión, que aprendieron.

Recomendaciones:

- Implementa funciones y has un uso adecuado de parámetros.
- Otorga nombres significativos a tus variables.
- Que tu programa principal sea lo más simple posible.
- Incluye comentarios.
- Ve guardando versiones de tu programa, si algo ya funcionó guarda esa versión y trabaja con una copia, de manera que siempre puedas volver a la versión anterior funcional.
- Prueba tu código para muchos casos.
- Que tu código tenga la indentación correcta.
- Elige bien tus estructuras de decisión o de repetición de acuerdo con el caso.