



ISIS-1221

INTRODUCCIÓN A LA PROGRAMACIÓN

Nivel 3 – Laboratorio lectura de archivos y recorrido de diccionarios

Objetivos

1. Practicar la lectura de archivos con formato CSV.
2. Ejercitar la implementación de algoritmos de recorrido de diccionarios.
3. Familiarizarse con la estructura compleja de datos de diccionarios dentro de diccionarios.
4. Fomentar la habilidad de descomponer un problema en subproblemas y de implementar funciones que los resuelven, lo que se conoce comúnmente como la técnica de “Dividir y Conquistar”.

Preparación

Cree una carpeta de trabajo y descargue allí el archivo [n3-14-esqueleto.zip](#) que se encuentra adjunto a este enunciado en BrightSpace. Descomprima este archivo y abra desde Spyder los archivos [librería_vuelos.py](#) e [interfaz_consola.py](#).

Instrucciones

Complete las funciones que se encuentran declaradas en el módulo [librería_vuelos.py](#) teniendo en cuenta la documentación y la información que viene a continuación.

Note que se le está entregando la interfaz basada en consola ([interfaz_consola.py](#)) completamente implementada. Abra el archivo y familiarícese con el código de las diferentes funciones: `iniciar_aplicacion`, `mostrar_menu` y `ejecutar_XX`.

Lea cuidadosamente el código de las funciones `ejecutar_vuelos_directos`, `ejecutar_vuelos_con_una_escalas` y `ejecutar_sugerir_aerolinea`. Asegúrese de que entendió cómo se valida el resultado de la función del módulo que es llamado en cada caso para mostrar al usuario el mensaje correspondiente. Si no entendió, pida explicación a la profesora o a los monitores.

Puede probar la implementación de sus funciones cargando la información que se encuentra en los archivos `“vuelos_cortos.csv”` y `“vuelos.csv”`. El primero tiene sólo 10 vuelos para que usted pueda probar rápidamente mientras que el segundo tiene la información de más de 7000 vuelos.

Problema 1: Vuelos directos

Complete la función `vuelos_directos` que recibe por parámetro un diccionario de vuelos, un aeropuerto de origen y un aeropuerto destino y retorna una lista de diccionarios con toda la información

(incluyendo los códigos) de todos los vuelos directos que van del aeropuerto de origen al aeropuerto de destino. Los diccionarios que se añaden en la lista deben tener como llave el código del vuelo y como valor el resto de la información del vuelo.

Si en el diccionario de vuelos no hay ningún vuelo para ese trayecto, la función debe retornar una lista vacía.

Problema 2: Vuelos con una escala

Complete la función `vuelos_con_una_escal` que recibe por parámetro un diccionario de vuelos, un aeropuerto de origen y un aeropuerto destino y retorna todos los vuelos con una escala que van del aeropuerto de origen al aeropuerto de destino. El retorno de esta función debe ser una lista de listas. Estas últimas son listas de dos elementos, que corresponden a los códigos de los 2 vuelos que permiten ir del origen al destino.

Si en el diccionario de vuelos no hay ningún vuelo para ese trayecto, la función debe retornar una lista vacía.

Problema 3: Sugerir aerolínea

Complete la función `sugerir_aerolinea` que recibe como parámetros un diccionario de vuelos, el código de un aeropuerto de origen y el código de un aeropuerto de destino y retorna el nombre de la aerolínea que se sugiere utilizar para ese trayecto. La aerolínea sugerida debe ser aquella que haya tenido el menor retraso promedio en ese trayecto. Si en el diccionario no hay ningún vuelo para ese trayecto, la función debe retornar `None`.

Entrega

Comprima los dos archivos modificados (consola y módulo) en un solo archivo .zip. El nombre del archivo comprimido tiene que seguir la siguiente estructura `n3-l4-login.zip`, donde `login` corresponde a su nombre de usuario de Uniandes. Por ejemplo, si su usuario es `p.perez10`, su archivo debe llamarse: `n3-l4-p.perez10.zip`. Entregue el archivo a través de BrightSpace en el laboratorio del nivel 3 designado como “L4: Lectura de archivos y recorrido de diccionarios”.