



ISIS-1221 INTRODUCCIÓN A LA PROGRAMACIÓN

Proyecto de Nivel 3 Avistamiento de OVNI

Objetivo general

El objetivo general de este proyecto es que usted practique todos los conceptos estudiados en el nivel 3 del curso. Recuerde que este proyecto debe realizarse de forma **completamente individual**.

Objetivos específicos

1. Practicar la lectura y escritura de archivos con formato CSV.
2. Ejercitar la implementación de algoritmos de recorrido y modificación de listas.
3. Familiarizarse con la estructura compleja de datos de diccionarios dentro de listas.
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”.
- 5.

Contexto

OVNI

Una de las preguntas que ha prevalecido dentro del pensamiento humano es si estamos solos en el universo, y aunque aún no se le ha dado respuesta, teniendo en cuenta la vasta amplitud del universo, la probabilidad de que exista alguna otra forma de vida inteligente en otros lugares es posible. ¿Y si otras civilizaciones más inteligentes ya nos visitaron? ¿Y si nos están mirando en este momento? ¿Y si los avistamientos desconocidos en el cielo están relacionados con esto? A lo largo de la historia, varias personas han manifestado el avistamiento de objetos voladores en el cielo. Dichos objetos, reciben el nombre de objeto volador no identificado (OVNI) refiriéndose a la observación de un objeto volador, real o aparente, que no puede ser identificado por el observador y cuyo origen sigue siendo desconocido después de realizar una investigación.

En abril del 2020, El Pentágono desclasificó tres videos, grabados por la Marina estadounidense, confirmando la autenticidad de estos objetos. En dichos videos, se observan OVNI



Figura 1. Ejemplo de un avistamiento de un ovni

Según estadísticas, se ha podido establecer que una de cada tres personas ha sido testigo de alguna situación que puede clasificar como un fenómeno OVNI. Lo anterior, vuelve a abrir los interrogantes expuestos anteriormente sobre la vida inteligente en el universo.

National UFO Reporting Center (NUFORC)

NUFORC es una organización de los Estados Unidos que investiga firmas de OVNI y/o contactos extraterrestres. Gracias a su arduo trabajo de recolección de datos ha creado una base de datos que contiene alrededor de 80.000 registros de avistamientos de OVNI alrededor del mundo durante el último siglo. Esta información podría ser de utilidad para encontrar una respuesta a la pregunta *¿Estamos solos en el universo?*

Descripción de la aplicación

En este proyecto se va a implementar una estructura para analizar los registros de avistamientos de OVNI sobre una porción de la base de datos de NUFORC.

El programa debe ser capaz de recibir los registros de avistamiento a partir de un archivo en formato CSV (Comma-Separated Values) que tiene 10 columnas:

- **Datetime:** Hora y fecha del avistamiento registrado en el formato “YYYY-MM-DD hh:mm:ss”.
- **City:** Ciudad en la que se registró el avistamiento.
- **State:** Estado en el que se registró el avistamiento.
- **Country:** País en el que se registró el avistamiento.
- **Shape:** Descripción de la forma del OVNI involucrado en el avistamiento.
- **Duration:** Duración en segundos del avistamiento registrado.
- **Comments:** Comentarios del avistamiento registrado.
- **Date posted:** Fecha de publicación del comentario.
- **Latitude:** Latitud de la zona geográfica donde sucedió el avistamiento registrado.
- **Longitude:** Longitud de la zona geográfica donde sucedió el avistamiento registrado.

NOTA: Tenga en cuenta que es posible que para algunos registros de avistamientos no se tenga certeza de todos los datos. Para estos casos, el valor de la columna es vacío “”.

Por ejemplo, el siguiente archivo sería válido para ser cargado dentro del sistema:

```
datetime,city,state,country,shape,duration (seconds),comments,date posted,latitude,longitude
1955-10-10 17:00:00,chester(uk/england),,gb,circle,20.0,Green/Orange circular disc over
Chester&#44 England,1/21/2008,53.2,-2.916667
1974-10-10 23:00:00,HUDSON,KS,US,LIGHT,1200.0,The light chased us.,7/25/2004,38.1055556,-
98.6597222
```

Para el manejo de los datos, se debe leer el archivo .csv y **desarrollar una estructura cuyo componente principal es un diccionario de países**, en el cual **las llaves sean la representación alfabética de cada país**, y **el valor una lista de diccionarios**. Cada diccionario de la lista contendrá la información de un avistamiento registrado en dicho país. Los diccionarios de cada avistamiento tendrán las llaves “datetime” con la fecha y hora del avistamiento en el formato YYYY-MM-DD hh:mm:ss, “city” con la ciudad del avistamiento, “state” con el estado del avistamiento, “shape” con la forma del OVNI observado en el avistamiento, “duration” con la duración en segundos del avistamiento, “comments” con los comentarios relacionados con el avistamiento, “date posted” con la fecha de publicación del avistamiento, “latitude” con la latitud geográfica del avistamiento y “longitude” con la longitud geográfica del avistamiento. A continuación, se muestra la estructura deseada para el manejo de los datos:

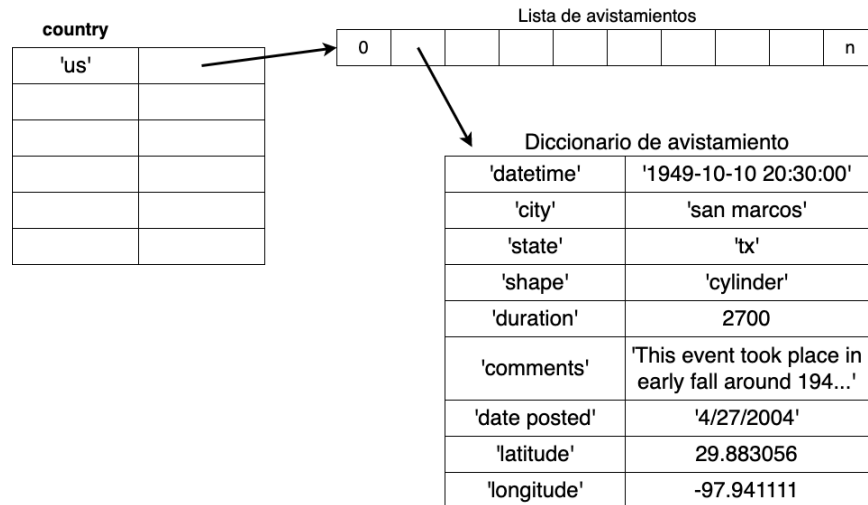


Figura 2. Estructura deseada para el manejo de los datos del archivo .csv

La aplicación debe permitir al usuario ejecutar las siguientes acciones:

1. Cargar un archivo con la información de los avistamientos de la base de datos del NUFORC.
2. Buscar los avistamientos por fecha.
3. Buscar avistamientos por ciudad.
4. Determinar el país con el mayor número de avistamientos.
5. Consultar los avistamientos dentro de un rango de fechas.
6. Consultar los avistamientos en una zona geográfica dentro de un radio determinado.
7. Contar los avistamientos registrados por cada estado.
8. Consultar el avistamiento más largo y el más corto por cada país.
9. Contar el número de avistamientos por la forma del OVNI.
10. Consultar los avistamientos con una duración mayor a una cantidad determinada.
11. Consultar los avistamientos que contengan una cadena dentro de sus comentarios.

Actividad 0: Preparación del ambiente de trabajo

1. Cree una carpeta para trabajar, poniéndole su nombre o login.
2. Descargue de Bloque Neón el archivo con el “esqueleto” del proyecto (n3-ovnis-esqueleto.zip) y descomprímalo en su carpeta de trabajo. El esqueleto consiste en un conjunto de archivos que usted va a usar o a modificar.
3. Abra Spyder y cambie la carpeta de trabajo para que sea la carpeta con el esqueleto.

Actividad 1: Construir el módulo de funciones

Usando Spyder, cree en su carpeta de trabajo un nuevo archivo con el nombre “ovnis.py”. En este archivo usted va a construir el módulo en el que va a implementar las funciones que responden a los requerimientos de la aplicación. **Defina, documente e implemente** las funciones descritas a continuación en su nuevo archivo. **Lea cuidadosamente las descripciones de las funciones para determinar los parámetros de entrada junto con sus tipos y el valor de retorno con su respectivo tipo.**

Función 1:

Implemente una función que reciba como parámetro el nombre de un archivo que contiene la información de los avistamientos de la base de datos de NUFORC y la cargue en el programa bajo la forma de **un diccionario de listas de diccionarios**. Cada avistamiento se va a representar utilizando un diccionario que tiene las siguientes llaves:

Llave	Tipo de dato	Descripción de la llave
<i>datetime</i>	str	Hora del avistamiento en el formato YYYY-MM-DD hh:mm:ss.
<i>city</i>	str	Ciudad donde se registró el avistamiento.
<i>state</i>	str	Estado donde se registró el avistamiento.
<i>shape</i>	str	Forma del OVNI involucrado en el avistamiento.
<i>duration</i>	float	Duración en segundos del avistamiento.
<i>comments</i>	str	Comentarios relacionados con el avistamiento.
<i>date posted</i>	str	Fecha de publicación del avistamiento.
<i>latitude</i>	float	Latitud geográfica del avistamiento.
<i>longitude</i>	float	Longitud geográfica del avistamiento.

Esta función debe retornar el diccionario que tiene como llave cada país y como valor asociado su respectiva lista de avistamientos (de diccionarios) creada. La estructura objetivo se muestra en la Figura 2.

Función 2:

Implemente una función que reciba por parámetro el diccionario completo de avistamientos y una fecha en el formato “YYYY-MM-DD” y retorne una lista con los avistamientos que fueron registrados en la fecha ingresada por parámetro.

Función 3:

Implemente una función que reciba por parámetro el diccionario completo de avistamientos y retorne un diccionario que tenga como llaves los nombres de cada ciudad y como valor asociado una lista con los avistamientos registrados en dicha ciudad. Tenga en cuenta que pueden existir ciudades homónimas –que comparten el mismo nombre pero que son ciudades distintas– que se agreguen al valor de cada llave. Haga caso omiso a esas ciudades homónimas y agréguelas a los avistamientos con ese mismo nombre de ciudad.

Función 4:

Implemente una función que reciba por parámetro el diccionario completo de avistamientos y un número de segundos, y retorne un diccionario con la misma estructura del original, pero solo con los avistamientos que tienen una duración mayor a la ingresada por parámetro.

Función 5:

Implemente una función que reciba por parámetro el diccionario completo de avistamientos, una fecha de inicio y una fecha de fin y retorne una lista con todos los avistamientos que fueron registrados dentro del rango de fechas ingresado por parámetro. Las fechas ingresadas deben estar en el formato “YYYY-MM-DD”.

Función 6:

Implemente una función que reciba por parámetro el diccionario completo de avistamientos, una latitud y una longitud geográfica y un radio determinado y retorne una lista con los avistamientos que se encuentran dentro del radio que tiene como centro la longitud y la latitud ingresada por parámetro. Tenga en cuenta que para determinar si un avistamiento está dentro del radio se debe determinar la distancia entre la latitud y la longitud ingresada por parámetro y la latitud y longitud registrada en el avistamiento, si la distancia es menor al radio entonces el avistamiento se encuentra dentro del radio estipulado.

Nota: Para este requerimiento, se entrega a continuación la función “distancia_entre_dos_puntos”, la cual toma dos longitudes y dos latitudes y retorna la distancia entre los puntos geográficos.

```
from math import radians, cos, sin, asin, sqrt
def distancia_entre_dos_puntos(lat1: float, lon1: float, lat2: float, lon2: float)
-> float:

    lon1 = radians(lon1)
    lon2 = radians(lon2)
    lat1 = radians(lat1)
    lat2 = radians(lat2)

    dif_lon = lon2 - lon1
    dif_lat = lat2 - lat1
    a = sin(dif_lat / 2)**2 + cos(lat1) * cos(lat2) * sin(dif_lon / 2)**2
    c = 2 * asin(sqrt(a))
    return c * 6371
```

Función 7:

Implemente una función que reciba por parámetro el diccionario completo de avistamientos, una cantidad de avistamientos, y un mes y año en el formato ‘YYYY-MM’, e indique si en ese mes hubo por lo menos tantos avistamientos como indique la cantidad ingresada por parámetro. La función debe retornar un valor booleano que indique si hubo o no la cantidad mínima de avistamientos durante el mes dado.

Función 8:

Implemente una función que reciba por parámetro el diccionario completo de avistamientos y un país, y retorne un diccionario con dos llaves: “corto” y “largo”. La llave “corto” debe tener como valor el avistamiento registrado con menos duración y la llave “largo” debe tener el avistamiento con mayor duración registrado en el archivo, para el país dado como parámetro.

Función 9:

Implemente una función que reciba por parámetro el diccionario completo de avistamientos y retorne un diccionario con el conteo de avistamientos por forma. El diccionario esperado debe tener como llaves las diferentes formas registradas en la base de datos y como valor, el número de avistamientos registrados con la forma correspondiente.

Función 10:

Implemente una función que reciba por parámetro el diccionario de avistamientos y una duración en segundos, y retorne un diccionario con las llaves “pais” y “avistamientos”, y tengan como valor el nombre del país con el mayor número de avistamientos registrados y el número de avistamientos de dicho país, entre los avistamientos que cumplan con una duración mayor a la pasada por parámetro. La función 4 puede serle de utilidad para resolver este requerimiento.

Función 11:

Implemente una función que reciba por parámetro el diccionario completo de avistamientos y una cadena de caracteres, y retorne una lista con todos los avistamientos que contienen la cadena ingresada dentro de los comentarios del avistamiento registrado.

Actividad 2: Completar la interfaz de usuario basada en consola

1. En esta actividad usted tiene que construir la interfaz basada en consola para que el usuario interactúe con la aplicación. Para construir esta interfaz usted debe completar el archivo `consola_ovnis.py`, la cual ya tiene una parte implementada que le facilitará su trabajo. Usted debe modificar los elementos marcados con la etiqueta `TODO`.
2. Pruebe la interfaz basada en consola ejecutando el archivo `"consola_ovnis.py"`. Verifique que las funcionalidades de su aplicación se comporten de acuerdo con lo esperado.
3. Pruebe cargando el archivo `ovnis_clean.csv` o cree su propio archivo de prueba respetando el mismo formato.

Entrega

1. Comprima la carpeta llamada "esqueleto" con su proyecto resuelto. El archivo debe llamarse **N3-PROY-login.zip**, donde login es su nombre de usuario de Uniandes.
2. Entregue el archivo comprimido a través de Bloque Neón en la actividad designada como **Proyecto del Nivel 3**.