



En este proyecto se pretende trabajar con un fichero (dataset) que contiene información sobre determinados avistamientos de ovnis. Los datos se encuentran almacenados en el fichero en formato CSV codificado en UTF-8.

Construya un proyecto denominado “**Avistamientos**” con las carpetas habituales:

- **src**: que contendrá los módulos **ovni.py** y **test_ovni.py**
- **data**: que contendrá el archivo **avistamiento.csv**
- **doc**: en el que copiará este **enunciado**

Cada registro del fichero ocupa una línea y contiene los datos correspondientes a un avistamiento con los siguientes campos:

- Fecha y hora
- Ciudad
- Acrónimo del estado
- Forma del objeto observado
- Duración en segundos
- Una descripción textual del avistamiento
- Latitud del lugar donde se produjo el avistamiento
- Longitud del lugar donde se produjo el avistamiento

A continuación, se muestran las 5 primeras líneas del fichero (acortando la descripción del avistamiento), teniendo en cuenta que la primera línea es una cabecera que contiene los nombres de los campos.

```
date_time,city,estate,forma,duracion,comments,latitud,longitud
07/04/2011 22:00,muncie,in,light,240, ((HOAX??)) 4th of July ufo...,40.1933333,-85.3863889
04/07/2005 17:01,deming (somewhere near),nm,changing,1200, ((NUFORC...,32.2686111,-107.7580556
03/12/2010 19:56,erie,pa,changing,300, 3/12/10Viewed a comet like...,42.1291667,-80.0852778
07/04/2013 22:25,seattle,wa,unknown,600, A RED Light was seen over...,47.6063889,-122.3308333
```

El proyecto tiene 4 grupos de ejercicios:

- Carga de datos
- Operaciones de filtrado, conteo y suma
- Operaciones con máximos, mínimos y ordenación
- Operaciones con diccionarios

Carga de datos

1. Función de lectura de datos

lee_avistamientos:

Función encargada de leer el fichero con los avistamientos y devolver una lista de tuplas de tipo una estructura de datos en memoria, que será una lista de tuplas de tipo “**Avistamiento**” (debe definir su propia namedtuple aunque los campos latitud y longitud debe estar agrupadas en un campo de tipo “**Coordenada**”).

- ENTRADA: ruta del fichero csv que contiene los datos: →str
- SALIDA: [Avistamiento(datetime, str, str, str, str, int, str, Coordenada(float, float))]

El resultado esperado es:

Número de registros leídos: 31682

Los 2 primeros registros: [avistamiento(fechahora=datetime.datetime(2011, 7, 4, 22, 0), ciudad='muncie', estado='in', forma='light', duración=240, comentarios='((HOAX??)) 4th of July ufo and a image of a ghostly alien face very disturbing.', coordenada=coord(latitud=40.1933333, longitud=-85.3863889)), avistamiento(fechahora=datetime.datetime(2005, 4, 7, 17, 1), ciudad='deming (somewhere near)', estado='nm', forma='changing', duración=1200, comentarios='((NUFORC Note: Helium-filled heliostat.



PD)) The shape-shifting object danced around at various speed height and direction.',
coordenada=coord(latitud=32.2686111, longitud=-107.7580556))]
Los 2 últimos registros: [avistamiento(fechahora=datetime.datetime(2008, 3, 31, 0, 0), ciudad='eagle river', estado='ak', forma='light', duración=1200, comentarios='Zig-zaging light moves over north horizon of Alaska town.', coordenada=coord(latitud=61.3213889, longitud=-149.5677778)),
avistamiento(fechahora=datetime.datetime(2003, 12, 5, 11, 0), ciudad='stafford', estado='va', forma='light', duración=300, comentarios='Zipping White Light.', coordenada=coord(latitud=38.4219444, longitud=-77.4086111))]

Operaciones de filtrado, conteo y suma

2. **número_avistamientos_en_una_fecha:**

Función que dada una lista de tuplas de tipo avistamiento y una fecha (sin tener en cuenta la hora), obtiene el número total de avistamientos que se han producido en dicha fecha.

- ENTRADA: → [Avistamiento(datetime, str, str, str, str, int, str, Coordenada(float, float))]
→ datetime.date
- SALIDA: → int

El resultado esperado para el 26 de diciembre de 2011 es:

El día 2011-12-26 se produjeron 7 avistamientos

3. **número_diferentes_formas_en_estados:**

Función que dada una lista de tuplas de tipo avistamiento y un conjunto con las denominaciones de varios estados, obtiene el número total de diferentes formas observadas en dicho conjunto de estados.

- ENTRADA: → [Avistamiento(datetime, str, str, str, str, int, str, Coordenada(float, float))]
→ {str}
- SALIDA: → int

El resultado esperado para probar con los siguientes estados "nm, pa, in, wa":

Se observaron 23 diferentes formas en los estados: {'nm', 'in', 'pa', 'wa'}

4. **duración_total:**

Función que dada una lista de tuplas de tipo avistamiento y un estado, obtiene el total de segundos que han durado los avistamientos en dicho estado.

- ENTRADA: → [Avistamiento(datetime, str, str, str, str, int, str, Coordenada(float, float))]
→ str
- SALIDA: → int

El resultado esperado para el estado "in". También para probar con "wa"

Duración total de los avistamientos en in: 3,318,305 segundos

Duración total de los avistamientos en wa: 1,822,712 segundos

5. **avistamientos_cercanos_coordenada:**

Función que dada una lista de tuplas de tipo avistamiento, una coordenada y una distancia devuelva una lista de tuplas con la fecha (sin la hora), la ciudad, el estado y la coordenada de los avistamientos observados a una distancia menor que la dada. **NOTA.**- cree una función auxiliar "**distancia**" que dada dos tuplas de tipo Coordenadas, devuelva la distancia euclídea entre ellas.

- ENTRADA: → [Avistamiento(datetime, str, str, str, str, int, str, Coordenada(float, float))]
→ Coordenada(float, float)
→ float
- SALIDA: → [(date, str, str, Coordenada(float, float))]



Resultados esperados para la coordenada (38.26, -77.18) y radio 0.25

```
Avistamientos cercanos a coord(latitud=38.26, longitud=-77.18) a menos de 0.25:  
[(datetime.date(1993, 6, 22), 'colonial beach', 'va', coord(latitud=38.2544444, longitud=-76.9638889)),  
(datetime.date(2013, 10, 24), 'dahlgren', 'va', coord(latitud=38.3311111, longitud=-77.0513889)),  
(datetime.date(2008, 10, 19), 'king george', 'va', coord(latitud=38.2680556, longitud=-77.1847222))]
```

Operaciones con máximos, mínimos y ordenación

6. avistamiento_mayor_duración:

Función que dada una lista de tuplas de tipo avistamiento y una forma, devuelva una tupla con la ciudad, la fecha y hora y la duración del avistamiento con mayor duración de los observados con la forma dada.

- ENTRADA: → [Avistamiento (datetime, str, str, str, str, int, str, Coordenada(float, float))]
→ str
- SALIDA: → (str, datetime, int)

Resultado esperado para la forma "circle":

```
El avistamiento en de forma circle de mayor duración es: (datetime.datetime(1984, 3, 15, 20, 0), 'griffin',  
7894800)
```

7. avistamientos_entre_fechas:

Función que dada una lista de tuplas de tipo avistamiento y dos fechas (sin horas) que se suponen bien ordenadas (la primera menor o igual que la segunda), devuelva una lista de tuplas de tipo avistamiento ordenadas de más reciente a más antigua que se hayan producido entre las fechas dadas (ambas incluidas).

NOTA.- Las fechas pueden ser parámetros que se omitan, en cuyo caso tomarán el valor None y no existirá filtro para la/s fecha/s de que se trate/n.

- ENTRADA: → [Avistamiento (datetime, str, str, str, str, int, str, Coordenada(float, float))]
→ date (*puede omitirse*)
→ date (*puede omitirse*)
- SALIDA: → [Avistamiento (datetime, str, str, str, str, int, str, Coordenada(float, float))]

Resultados esperados para avistamientos entre el 1/5/2005 y 2/5/2005:

```
Avistamientos entre el 01 de May de 2005 y el 02 de May de 2005:  
---> avistamiento(fechahora=datetime.datetime(2005, 5, 2, 21, 0), ciudad='clovis', estado='ca',  
forma='light', duración=60, comentarios='Bright white light flased in the sky followed by 2 red flashing  
lights move up and down back and forth.', coordenada=coord(latitud=36.8252778, longitud=-  
119.7019444))  
---> avistamiento(fechahora=datetime.datetime(2005, 5, 2, 14, 5), ciudad='palmdale', estado='ca',  
forma='circle', duración=600, comentarios='Two (2) silver bright round disk(s) that touched then  
seperated numerous times seem to follow chemtrail over Palmdale California.',  
coordenada=coord(latitud=34.5794444, longitud=-118.1155556))  
---> avistamiento(fechahora=datetime.datetime(2005, 5, 2, 0, 30), ciudad='angels camp', estado='ca',  
forma='circle', duración=120, comentarios='Strange Object in Angels Camp CA. ((NUFORC Note:  
Probable sighting of the star Sirius we suspect. PD))', coordenada=coord(latitud=38.0683333, longitud=-  
120.5386111))  
---> avistamiento(fechahora=datetime.datetime(2005, 5, 1, 23, 0), ciudad='palmdale', estado='ca',  
forma='sphere', duración=300, comentarios='dancing light above lancaster',  
coordenada=coord(latitud=34.5794444, longitud=-118.1155556))  
---> avistamiento(fechahora=datetime.datetime(2005, 5, 1, 16, 11), ciudad='fallbrook', estado='ca',  
forma='changing', duración=30, comentarios='Huge bright liigt', coordenada=coord(latitud=33.3763889,  
longitud=-117.2502778))
```



```
---> avistamiento(fechahora=datetime.datetime(2005, 5, 1, 6, 20), ciudad='independence', estado='ky',
forma='other', duración=300, comentarios='stationary disk-like object seen near I-75/Independence KY',
coordenada=coord(latitud=38.9430556, longitud=-84.5441667))
---> avistamiento(fechahora=datetime.datetime(2005, 5, 1, 1, 0), ciudad='portland', estado='or',
forma='disk', duración=300, comentarios='Orange disk-like orbs over Downtown Portland. ((NUFORC
Note: Student report. PD))', coordenada=coord(latitud=45.5236111, longitud=-122.675))
---> avistamiento(fechahora=datetime.datetime(2005, 5, 1, 1, 0), ciudad='portland', estado='or',
forma='oval', duración=420, comentarios='Three amber/orange objects patrol Portland skyline',
coordenada=coord(latitud=45.5236111, longitud=-122.675))
```

8. avistamiento_comentario_más_largo_del_año:

Función que dada una lista de tuplas de tipo avistamiento, una palabra (no debe importar que sean mayúsculas o minúsculas) y un año (que se puede omitir y en dicho caso tomará el valor None y no habrá que filtrar por el año), devuelva una tupla de tipo avistamiento con el comentario más largo que contenga la palabra dada del año dado.

- ENTRADA: →[Avistamiento (datetime, str, str, str, str, int, str, Coordenada(float, float))]
 →str
 →int (*puede omitirse*)
- SALIDA: →Avistamiento (datetime, str, str, str, str, int, str, Coordenada(float, float))

Resultados esperados para avistamientos que contenga la palabra "Object". Hay dos ejecuciones una sin año y otra con el 2009

Avistamiento con la palabra "Object" de cualquier año con el comentario más largo:
avistamiento(fechahora=datetime.datetime(2007, 10, 25, 22, 30), ciudad='marion', estado='ia',
forma='light', duración=900, comentarios="\t\t \tl was outside smoking a cigarette. I looked over to the
field across the street. I saw the first object and thought maybe it's jus",
coordenada=coord(latitud=42.0341667, longitud=-91.5975))

Avistamiento con la palabra "Object" del año 2009 con el comentario más largo:
avistamiento(fechahora=datetime.datetime(2009, 9, 28, 22, 0), ciudad='green bay', estado='wi',
forma='triangle', duración=300, comentarios='09-28-09 Green Bay WI Triangle Shape Object 3 Lights on
one side One Light on the bottom side on the other and red light in center.',
coordenada=coord(latitud=44.5191667, longitud=-88.0197222))

Operaciones con diccionarios "sencillos"

9. duración_total_por_fechas:

Función que dada una lista de tuplas de tipo avistamiento, devuelva un diccionario que a cada fecha le haga corresponder la suma de las duraciones de los avistamientos que corresponden a cada fecha.

- ENTRADA: →[Avistamiento (datetime, str, str, str, str, int, str, Coordenada(float, float))]
- SALIDA: →{date: int}

Resultados esperados: Dado que se visualizan muchos datos que no caben en la consola, se le sugiere que en el test convierta el diccionario en una lista, la ordene por el primer elemento (con contendrá la fecha) y visualice los 5 últimos elementos

Duración por fechas:
[(datetime.date(2014, 5, 4), 1815),
(datetime.date(2014, 5, 5), 605),
(datetime.date(2014, 5, 6), 3962),
(datetime.date(2014, 5, 7), 1021),



```
(datetime.date(2014, 5, 8), 900))
```

10. avistamientos_por_fechas:

Función que dada una lista de tuplas de tipo avistamiento, devuelva un diccionario que a cada fecha le haga corresponder un conjunto con los avistamientos que corresponden a cada fecha.

- ENTRADA: → [Avistamiento (datetime, str, str, str, str, int, str, Coordenada(float, float))]
- SALIDA: → {date: {Avistamiento (datetime, str, str, str, str, int, str, Coordenada(float, float))}}

Resultados esperados: Dado que se visualizan muchos datos que no caben en la consola, se le sugiere que en el test convierta el diccionario en una lista, la ordene por el primer elemento (con contendrá la fecha) y visualice los dos últimos elementos

Avistamientos por fechas:

```
[(datetime.date(2014, 5, 7), {avistamiento(fechahora=datetime.datetime(2014, 5, 7, 20, 30),
ciudad='seagrove beach', estado='fl', forma='fireball', duración=60, comentarios='Around 8:30 pm I was
on the balcony of our condo and saw two fireball type items in the sky east over Panama City. There was
no noise a', coordenada=coord(latitud=30.3183333, longitud=-86.1302778)),
avistamiento(fechahora=datetime.datetime(2014, 5, 7, 15, 0), ciudad='san jose', estado='ca', forma='disk',
duración=30, comentarios='Black huge wobbling object disk shape going across sky in clouds',
coordenada=coord(latitud=37.3394444, longitud=-121.8938889)),
avistamiento(fechahora=datetime.datetime(2014, 5, 7, 21, 0), ciudad='theodore', estado='al',
forma='circle', duración=120, comentarios='Driving home and entered my neighborhood and saw 2
blinking lights in sky to close to be airplanes. Then later was outside with 2 oth',
coordenada=coord(latitud=30.5475, longitud=-88.1752778)),
avistamiento(fechahora=datetime.datetime(2014, 5, 7, 3, 30), ciudad='calera', estado='al', forma='circle',
duración=240, comentarios='Orange/red sphere with blue or green outline or ring. Moved very slowly to
the west then disappeared.', coordenada=coord(latitud=33.1027778, longitud=-86.7536111)),
avistamiento(fechahora=datetime.datetime(2014, 5, 7, 9, 30), ciudad='port richy', estado='fl',
forma='unknown', duración=1, comentarios='Sonic boom denied by military source per McDill AFB Lt
Patrick Gargan. Reports by 911 from Holliday to Spring Hill Fl (about 40 miles)',
coordenada=coord(latitud=28.2713889, longitud=-82.7197222)),
avistamiento(fechahora=datetime.datetime(2014, 5, 7, 0, 0), ciudad='detroit', estado='mi',
forma='fireball', duración=180, comentarios='Fire balls in detroit sky.',
coordenada=coord(latitud=42.3313889, longitud=-83.0458333)),
avistamiento(fechahora=datetime.datetime(2014, 5, 7, 21, 10), ciudad='naugatuck', estado='ct',
forma='unknown', duración=300, comentarios='Large formation of lights in western sky flying north to
south...', coordenada=coord(latitud=41.4858333, longitud=-73.0511111)),
avistamiento(fechahora=datetime.datetime(2014, 5, 7, 17, 30), ciudad='lakeville', estado='mn',
forma='oval', duración=90, comentarios='((HOAX??)) Oval seen emitting haze and later smoke. Also took
2 pictures.', coordenada=coord(latitud=44.6497222, longitud=-93.2425))}),
(datetime.date(2014, 5, 8), {avistamiento(fechahora=datetime.datetime(2014, 5, 8, 0, 0),
ciudad='memphis', estado='tn', forma='rectangle', duración=900, comentarios='Standing at my window
around 0:00 brilliantly lit objects the size of aircraft carriers passed by one by one.',
coordenada=coord(latitud=35.1494444, longitud=-90.0488889))})])
```

11. formas_por_mes:

Función que dada una lista de tuplas de tipo avistamiento, devuelva un diccionario que a cada mes le haga corresponder las diferentes formas de los avistamientos que corresponden a cada mes.

- ENTRADA: → [Avistamiento (datetime, str, str, str, str, int, str, Coordenada(float, float))]
- SALIDA: → {str: {str}}



Ayuda: Si observa los resultados esperados las claves están en español, por lo que se le sugiere que cree una lista con los nombres de los meses del año en español y utilizando la propiedad **"month"** de un tipo date obtendrá el número de mes y podrá obtener utilizando la lista el nombre del mes como clave, pero recuerde que enero es el número 1 y el primer elemento de una lista comienza con el elemento 0.

Resultado esperado:

Formas por mes:

```
{'Julio': {'formation', 'oval', 'egg', 'chevron', 'delta', 'fireball', 'circle', 'cylinder', 'triangle', 'disk', 'teardrop', 'cigar', 'diamond', 'sphere', 'unknown', 'light', 'rectangle', 'changing', 'cross', 'flash', 'other', 'cone'}, 'Abril': {'formation', 'oval', 'egg', 'chevron', 'fireball', 'circle', 'cylinder', 'triangle', 'disk', 'teardrop', 'cigar', 'diamond', 'sphere', 'unknown', 'light', 'changing', 'rectangle', 'cross', 'flash', 'other', 'cone'}, 'Marzo': {'formation', 'oval', 'egg', 'chevron', 'fireball', 'circle', 'cylinder', 'triangle', 'teardrop', 'disk', 'cigar', 'diamond', 'sphere', 'unknown', 'light', 'changing', 'rectangle', 'cross', 'flash', 'other', 'cone'}, 'Septiembre': {'formation', 'oval', 'egg', 'chevron', 'fireball', 'circle', 'cylinder', 'triangle', 'disk', 'teardrop', 'diamond', 'cigar', 'sphere', 'unknown', 'light', 'changing', 'rectangle', 'cross', 'flash', 'other', 'cone'}, 'Junio': {'formation', 'oval', 'egg', 'chevron', 'fireball', 'circle', 'cylinder', 'triangle', 'disk', 'teardrop', 'cigar', 'diamond', 'sphere', 'unknown', 'light', 'rectangle', 'changing', 'cross', 'flash', 'other', 'cone'}, 'Mayo': {'formation', 'oval', 'egg', 'chevron', 'fireball', 'circle', 'cylinder', 'triangle', 'disk', 'teardrop', 'diamond', 'cigar', 'sphere', 'unknown', 'light', 'changing', 'rectangle', 'cross', 'flash', 'other', 'cone'}, 'Febrero': {'formation', 'oval', 'egg', 'chevron', 'delta', 'fireball', 'circle', 'cylinder', 'triangle', 'disk', 'teardrop', 'diamond', 'cigar', 'sphere', 'unknown', 'light', 'rectangle', 'changing', 'cross', 'flash', 'other', 'cone'}, 'Octubre': {'formation', 'oval', 'egg', 'chevron', 'fireball', 'circle', 'cylinder', 'triangle', 'disk', 'teardrop', 'diamond', 'cigar', 'sphere', 'unknown', 'light', 'changing', 'rectangle', 'cross', 'flash', 'other', 'cone'}, 'Noviembre': {'formation', 'oval', 'egg', 'chevron', 'delta', 'fireball', 'circle', 'cylinder', 'triangle', 'disk', 'teardrop', 'cigar', 'diamond', 'sphere', 'unknown', 'light', 'round', 'changing', 'rectangle', 'cross', 'flash', 'other', 'cone'}, 'Enero': {'formation', 'oval', 'egg', 'chevron', 'fireball', 'circle', 'cylinder', 'triangle', 'teardrop', 'disk', 'cigar', 'diamond', 'sphere', 'unknown', 'light', 'changing', 'rectangle', 'cross', 'flash', 'other', 'cone'}, 'Diciembre': {'formation', 'oval', 'egg', 'chevron', 'fireball', 'circle', 'cylinder', 'triangle', 'disk', 'teardrop', 'diamond', 'cigar', 'sphere', 'unknown', 'light', 'changing', 'rectangle', 'cross', 'flash', 'other', 'cone'}, 'Agosto': {'formation', 'oval', 'egg', 'chevron', 'fireball', 'circle', 'cylinder', 'triangle', 'disk', 'teardrop', 'cigar', 'diamond', 'sphere', 'unknown', 'light', 'rectangle', 'changing', 'cross', 'flash', 'other', 'cone'}}
```

12. número_de_avistamientos_por_mes:

Función que dada una lista de tuplas de tipo avistamiento, devuelva un diccionario que a cada mes le haga corresponder el número de los avistamientos que corresponden a cada mes.

- ENTRADA: → [Avistamiento (datetime, str, str, str, str, int, str, Coordenada(float, float))]
- SALIDA: → {str: {int}}

Resultados esperados (utilice la ayuda del ejercicio anterior)

Número de avistamientos por mes:

```
{'Julio': 3782, 'Abril': 2126, 'Marzo': 2132, 'Septiembre': 2989, 'Junio': 3129, 'Mayo': 2082, 'Febrero': 1825, 'Octubre': 3089, 'Noviembre': 2756, 'Enero': 2185, 'Diciembre': 2251, 'Agosto': 3336}
```

13. coordenadas más avistamientos:

Función que dada una lista de tuplas de tipo avistamiento, devuelva una tupla con la latitud y la longitud en que ha habido más avistamientos. A los efectos de este ejercicio, se considerará únicamente las partes enteras de las coordenadas. **Ayuda:** busque primero cuantos avistamientos hay en cada coordenada.

- ENTRADA: → [Avistamiento (datetime, str, str, str, str, int, str, Coordenada(float, float))]
- SALIDA: → (int, int)



Resultado esperado:

La coordenada con más avistamiento es: (47, -122)

14. horas_con_más_duración_avistamientos:

Función que dada una lista de tuplas de tipo avistamiento, con un conjunto de diferentes formas y un número “n” (que puede omitirse y en dicho caso “n” valdrá 3), devuelva una lista de “n” tuplas con las horas del día y la suma de las duraciones en dichas horas, en las que las sumas de las duraciones han sido mayores. A los efectos de este ejercicio, se considerará únicamente las partes enteras de las coordenadas.

- ENTRADA: → [Avistamiento (datetime, str, str, str, str, int, str, Coordenada(float, float))]
→ {str}
→ int (puede omitirse)
- SALIDA: → [(int,int)]

Resultados esperados para formas: 'circle', 'triangle' o 'cylinder' y n=5

Las 5 horas con más duración con formas {'circle', 'triangle', 'cylinder'}:
[(20, 8422203), (21, 1206068), (19, 1008863), (22, 903331), (23, 751470)]

Operaciones con diccionarios “complejos”

15. longitud_media_comentarios_por_estado:

Función que dada una lista de tuplas de tipo avistamiento, dos fechas (sin horas) que se suponen bien ordenadas (la primera menor o igual que la segunda), devuelva un diccionario en que a cada estado le haga corresponder el promedio de las longitudes de los comentarios de los avistamientos en dichos estados. **NOTA.**- Las fechas pueden ser parámetros que se omitan, en cuyo caso tomarán el valor None y no existirá filtro para la/s fecha/s de que se trate/n.

- ENTRADA: → [Avistamiento (datetime, str, str, str, str, int, str, Coordenada(float, float))]
→ date (puede omitirse)
→ date (puede omitirse)
- SALIDA: → {str:float}

Resultados esperados para avistamientos entre el 1 de enero de 2008 y el 31 de diciembre de 2009:

Longitud media de comentarios por estados entre 2008-01-01 y 2009-12-31:
{'oh': 75.58119658119658, 'nv': 74.14516129032258, 'ca': 77.96920289855072, 'va': 80.05333333333333, 'ne': 88.05882352941177, 'tx': 78.67918088737201, 'tn': 72.61428571428571, 'ri': 86.84615384615384, 'ky': 76.80645161290323, 'mn': 80.0625, 'fl': 73.96929824561404, 'nj': 79.325, 'il': 72.4724409448819, 'mo': 84.84615384615384, 'nc': 73.94791666666667, 'in': 80.29113924050633, 'ia': 71.09302325581395, 'wa': 74.3072625698324, 'vt': 72.3157894736842, 'az': 76.29032258064517, 'ga': 75.32911392405063, 'nh': 84.04761904761905, 'ks': 72.8, 'co': 75.84146341463415, 'ny': 75.32203389830508, 'mi': 73.92929292929293, 'md': 78.55555555555556, 'la': 65.16, 'me': 81.83870967741936, 'or': 80.07692307692308, 'al': 74.54285714285714, 'wi': 78.76271186440678, 'pa': 75.75925925925925, 'ma': 75.33333333333333, 'hi': 83.25, 'sc': 80.1842105263158, 'nm': 79.0, 'ct': 79.0, 'nd': 66.5, 'sd': 77.5, 'wv': 81.2, 'id': 90.25, 'ak': 83.83333333333333, 'ut': 83.2, 'ar': 68.45454545454545, 'ms': 74.25, 'ok': 78.92307692307692, 'mt': 81.0, 'de': 87.58333333333333, 'wy': 86.33333333333333}

16. porcentaje_avistamientos_por_forma:

Función que dada una lista de tuplas de tipo avistamiento, devuelva un diccionario en que a cada forma le haga corresponder el porcentaje en que aparecen dichas formas respecto al total de avistamientos.

- ENTRADA: → [Avistamiento (datetime, str, str, str, str, int, str, Coordenada(float, float))]
→ {str:float}
- SALIDA: → {str:float}

Resultados esperados:



Porcentaje según las formas:

```
{'light': 21.075058392778235, 'changing': 2.474591250552364, 'unknown': 7.240704500978474, 'triangle': 10.34025629695095, 'circle': 9.544851966416262, 'cone': 0.4040148980493656, 'other': 7.1270753109020895, 'disk': 6.464238368789849, 'cylinder': 1.5939650274603876, 'fireball': 8.048734297077205, 'cigar': 2.5945331734107695, 'sphere': 6.764093175935862, 'formation': 3.0995517959724763, 'oval': 4.942869768322707, 'egg': 0.8206552616627738, 'teardrop': 0.9942554131683606, 'chevron': 1.2877974875323528, 'diamond': 1.4393030743008648, 'rectangle': 1.7202196831008143, 'cross': 0.3093239063190455, 'flash': 1.7012814847547504, 'delta': 0.009469099173032005, 'round': 0.0031563663910106686}
```

17. porcentaje_avistamientos_por_forma_ordenada:

Función que dada una lista de tuplas de tipo avistamiento, devuelva una lista de tuplas, ordenadas alfabéticamente por la forma. Las tuplas estarán compuesta por la forma y el porcentaje en que aparecen dichas formas respecto al total de avistamientos.

- ENTRADA: → [Avistamiento (datetime, str, str, str, str, int, str, Coordenada(float, float))]
- SALIDA: → [(str:float)]

Resultados esperados:

Porcentaje según las formas:

```
[('changing', 2.474591250552364), ('chevron', 1.2877974875323528), ('cigar', 2.5945331734107695), ('circle', 9.544851966416262), ('cone', 0.4040148980493656), ('cross', 0.3093239063190455), ('cylinder', 1.5939650274603876), ('delta', 0.009469099173032005), ('diamond', 1.4393030743008648), ('disk', 6.464238368789849), ('egg', 0.8206552616627738), ('fireball', 8.048734297077205), ('flash', 1.7012814847547504), ('formation', 3.0995517959724763), ('light', 21.075058392778235), ('other', 7.1270753109020895), ('oval', 4.942869768322707), ('rectangle', 1.7202196831008143), ('round', 0.0031563663910106686), ('sphere', 6.764093175935862), ('teardrop', 0.9942554131683606), ('triangle', 10.34025629695095), ('unknown', 7.240704500978474)]
```

18. avistamientos_mayor_duración_por_estado:

Función que dada una lista de tuplas de tipo avistamiento y un límite “n” (que se puede omitir), devuelva un diccionario en que a cada estado le haga corresponder una lista de “n” tuplas como máximo, compuestas por la fecha y hora del avistamiento, la duración y la ciudad, de mayor duración para el estado de que se trate.

NOTA.- El límite “n” puede omitirse. En este caso deberá tomar el valor 3

- ENTRADA: → [Avistamiento (datetime, str, str, str, str, int, str, Coordenada(float, float))]
→ int (*puede omitirse*)
- SALIDA: → {str:[(datetime,int,str)]}

Resultados esperados para límite = 2: (se visualizan **sólo 4** estados)

Los 2 avistamientos de mayor duración en cada estado es:

```
{ 'in': [(datetime.datetime(2012, 4, 22, 21, 0), 2631600, 'cedar lake'), (datetime.datetime(1968, 5, 12, 9, 0), 109800, 'valparaiso')], 'nm': [(datetime.datetime(2011, 5, 4, 9, 0), 2102400, 'albuquerque'), (datetime.datetime(1945, 8, 16, 11, 30), 777600, 'san antonio')], 'pa': [(datetime.datetime(2002, 12, 8, 18, 45), 259200, 'sharpsville'), (datetime.datetime(2011, 6, 11, 20, 30), 37800, 'franklin')], 'wa': [(datetime.datetime(2006, 8, 19, 12, 0), 259200, 'trout lake'), (datetime.datetime(1995, 6, 13, 14, 0), 86400, 'sequim')], ...}
```

19. primer_avistamiento_por_forma:

Función que dada una lista de tuplas de tipo avistamiento y un estado (que se puede omitir), devuelva una lista de tuplas con el primer avistamiento de cada forma. Las tuplas deberán contener como primer elemento la forma y como segundo el avistamiento de que se trate. Asimismo, la lista de tuplas estará ordenada



alfabéticamente por las formas. **NOTA.**- El estado" puede omitirse. En este caso deberá tomar el valor None y no habrá filtro por este campo.

- ENTRADA: →[Avistamiento (datetime, str, str, str, str, int, str, Coordenada(float, float))]
→str (*puede omitirse*)
- SALIDA: →[(str, Avistamiento (datetime, str, str, str, str, int, str, Coordenada(float, float)))]

Resultado esperado para "wa": (se muestran **solo las 4 primeras** formas)

El primer avistamiento de wa ha sido:

```
[('changing', avistamiento(fechahora=datetime.datetime(1968, 6, 1, 22, 0), ciudad='renton', estado='wa',
forma='changing', duración=600, comentarios='A huge orange ball over Lake Hills in Renton WA that
turned into a triangle over Boeing field and disappeared.', coordenada=coord(latitud=47.4830556,
longitud=-122.2158333))), ('chevron', avistamiento(fechahora=datetime.datetime(1999, 6, 2, 21, 30),
ciudad='olympia', estado='wa', forma='chevron', duración=10, comentarios='Saw object flying very fast
noiseless south to north.', coordenada=coord(latitud=47.0380556, longitud=-122.8994444))), ('cigar',
avistamiento(fechahora=datetime.datetime(1961, 5, 1, 22, 0), ciudad='auburn', estado='wa',
forma='cigar', duración=300, comentarios='Grandmother saw a cigar shaped craft which she said was as
long as the valley is wide.', coordenada=coord(latitud=47.3075, longitud=-122.2272222))), ('circle',
avistamiento(fechahora=datetime.datetime(1952, 10, 15, 21, 0), ciudad='richland', estado='wa',
forma='circle', duración=5, comentarios='Night sighting of seven large luminous UFOs flying in V
formation', coordenada=coord(latitud=46.2858333, longitud=-119.2833333))),...]
```