FUNDAMENTOS DE PROGRAMACIÓN
Proyecto de Laboratorio: Centros Sanitarios
Autora: Toñi Reina. **Revisor:** Mariano González. **Última modificación:** 10/10/2022

PREPARACIÓN PREVIA

En este ejercicio vamos a trabajar con mapas, para lo que usaremos la librería folium[1]. Para instalar la librería, abra una ventana de comandos de Anaconda (Anaconda Prompt) y ejecute el siguiente comando:

```
conda install -c conda-forge folium
```

En este proyecto trabajaremos con datos proporcionados por la diputación de Cádiz sobre centros sanitarios de los municipios con población inferior a 50.000 habitantes de dicha provincia.

Los datos de los que partimos en el proyecto están en el fichero `centrosSanitarios.csv`. Si abre el fichero, comprobará que es un fichero en formato CSV (los datos están separados por punto y coma). Un extracto del mismo lo puede ver en la Figura 1.

```
NOMBRE;LOCALIDAD;LATITUD;LONGITUD;ESTADO;NUM_CAMAS;TIENE_ACCESO_DISCAPACITADOS;TIENE_UCI
CONSULTORIO ZAHARA DE LOS ATUNES;BARBATE;36.135051666002795;-5.843455923196172;BUENO;0;true;false
CENTRO DE SALUD BARBATE;BARBATE;36.189308321456515;-5.925475089376914;BUENO;0;true;false
CONSULTORIO JEDULA;ARCOS DE LA FRONTERA;36.72460010332263;-5.931456684613025;BUENO;0;true;false
```

Figura 1: Extracto del fichero centrosSanitarios.csv

Como puede observar, cada línea del fichero contiene el nombre de un centro sanitario, la localidad, la latitud, la longitud, el estado del centro, el número de camas que tiene, si tiene acceso para discapacitados y si tiene UCI (en ambos casos, el valor false representa que no tiene acceso o no tiene UCI, respectivamente).

En el proyecto tendremos que implementar algunas funciones que nos ayuden a explotar los datos de los que disponemos. El objetivo del ejercicio es que trabaje creando un proyecto desde cero, e implemente tanto las funciones como los tests para comprobar que las funciones trabajan como se espera. El proyecto creado debe tener la estructura que se muestra en la Figura 2, en la que hay tres módulos: `mapas.py`, que se le proporciona y solo tendrá que copiarlo en el proyecto; `coordenadas.py`, que tendrá que implementar desde cero y contendrá las funciones relacionadas con las coordenadas geográficas, y `centros.py`, que contendrá las funciones relacionadas con los centros sanitarios.

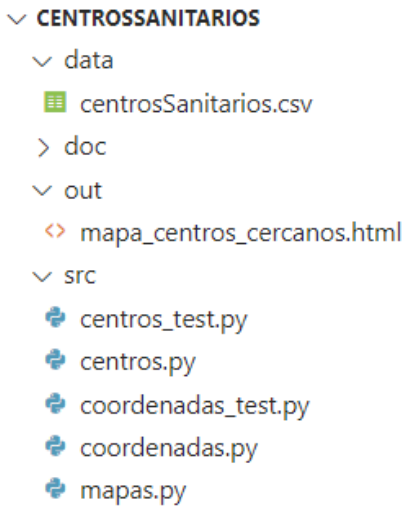

Figura 2: Estructura del proyecto

[1] https://python-visualization.github.io/folium/quickstart.html

Para gestionar las coordenadas en el módulo **coordenadas.py** se usará la siguiente definición de namedtuple:

```
Coordenada = namedtuple('Coordenada', 'latitud, longitud')
```

En dicho módulo, implemente las funciones que se especifican a continuación; además, una vez implementada una función, implemente su test correspondiente en el módulo **coordenadas_TEST.py** y compruebe que funciona como se espera.

1. **calcular_distancia**: recibe dos coordenadas de tipo Coordenada(float, float), y devuelve un float que representa la distancia euclídea entre esas dos coordenadas.
2. **calcular_media_coordenadas**: recibe una lista de Coordenada(float, float), y devuelve una Coordenada(float, float) cuya latitud es la media de las latitudes de la lista y cuya longitud es la media de las longitudes de la lista.

Para gestionar la información de los centros sanitarios se usará la siguiente definición de namedtuple en el módulo **centros.py**:

```
CentroSanitario = namedtuple('CentroSanitario', 'nombre, localidad, coordenada, longitud, estado, num_camas, acceso_minusvalidos, tiene_uci')
```

Implemente las funciones que se especifican a continuación en dicho módulo. Una vez implementada una función, implemente su test correspondiente en el módulo **centros_TEST.py** y compruebe que funciona como se espera.

1. **leer_centros**: recibe la ruta de un fichero CSV codificado en UTF-8, y devuelve una lista de tuplas de tipo CentroSanitario(str, str, Coordenada(float, float), str, int, bool, bool) conteniendo todos los datos almacenados en el fichero. Note que, aunque en el fichero la latitud y la longitud se almacenan de forma independiente, en la lista resultado deben almacenarse como una Coordenada(float, float).
2. **calcular_total_camas_centros_accesibles**: recibe una lista de tuplas de tipo CentroSanitario, y produce como salida un entero correspondiente al número total de camas de los centros sanitarios accesibles para discapacitados.
3. **obtener_centros_con_uci_cercanos_a**: recibe una lista de tuplas de tipo CentroSanitario, una tupla de tipo Coordenada que representa un punto, y un float que representa un umbral de distancia, y produce como salida una lista de tuplas (str, str, Coordenada(float, float)) con el nombre, la localidad y las coordenadas de los centros situados a una distancia de las coordenadas dadas como parámetro menor o igual que el umbral dado. Observe la Figura 3 para entender mejor el resultado de la función.

Figura 3: Radio de centros con UCI buscados

4. **generar_mapa**: recibe una lista de tuplas (str, str, Coordenada(float, float)) con el nombre, la localidad y las coordenadas de los centros, y una cadena que representa la ruta de un fichero html, y genera y graba en este fichero un mapa con los centros geolocalizados.

   Para implementar la función `generar_mapa` ayúdese de las funciones auxiliares que se implementan en el módulo `mapas.py`. Además, tenga en cuenta que:

   1. Primero debe crear un mapa. Use la media de las coordenadas de los centros para centrar el mapa.
   2. Después debe ir creando marcadores y añadiéndolos al mapa. Una vez creado el marcador, use `marcador.add_to(mapa)` para añadirlo al mapa.
   3. Una vez añadidos todos los marcadores, guarde el mapa en el archivo html con `mapa.save(fichero)`.

   El resultado deber ser un fichero con un mapa similar al de la Figura 4.

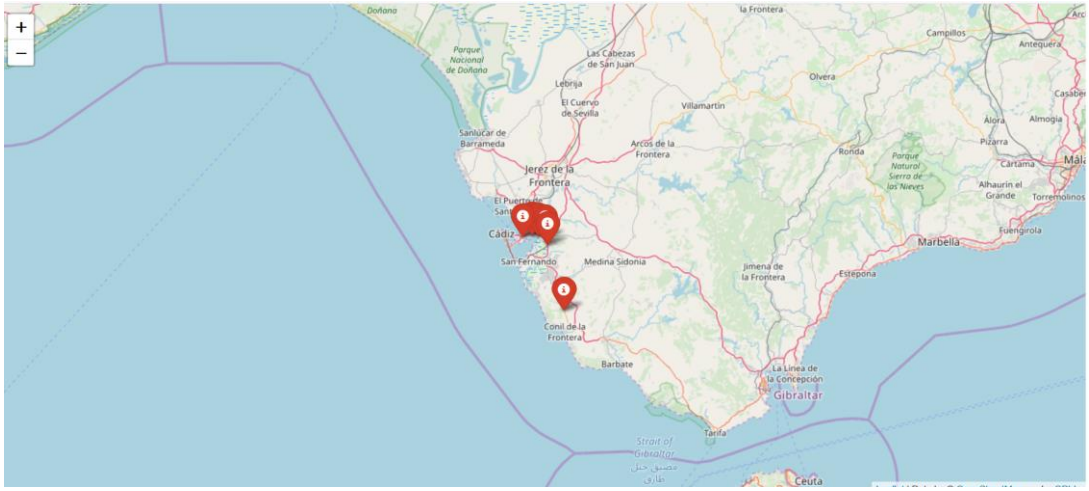

Figura 4: Mapa con centros de salud geolocalizados

RETO

La fórmula de la distancia euclídea empleada en este proyecto no es la más adecuada cuando se quieren calcular distancias entre dos puntos del globo terrestre, ya que no tiene en cuenta la curvatura de la tierra. Para calcular la distancia entre dos puntos del globo terrestre se usa una aproximación que viene dada por la fórmula de Haversine[1]. Implemente una función distancia_harvesine para que en el proyecto se hagan unos cálculos más realistas.

[1] https://es.wikipedia.org/wiki/F%C3%B3rmula_del_haversine