DOMINGO 30 ENERO.

En base a la búsqueda para la utilización de openstreetmap se llegaron a varias a proximaciones.

Dado que openstreet map es un proyecto opensource entonces muchas de las consultas no se hacen directamente con la API de esta, debido a esto en la documentacion de OSM nos recomienda consultar otras bases de datos o aplicaciones de terceros que ellos utilizan para consultar diferente tipos de datos. Se podria ver como un conjunto de fichas de lego que fueron armadas en diferentes instancias pero luego se pueden ir uniendo.

https://wiki.openstreetmap.org/wiki/Databases\_and\_data\_access\_APIs

Entonces para resolver el problema de buscar la distancia y el camino mas corto entre dos puntos o mas, esto no se puede resolver directamente con OSM, dado esto toca ir mas allá en la busqueda.

Luego de buscar se llego a una API que nos da la información requerida esta API es OSRM. (http://project-osrm.org/docs/v5.24.0/api/#). Exactamente utilizando el servicio route service. Un ejemplo de consulta seria la siguiente -> https://router.project-osrm.org/route/v1/car/-75.538971,6.345817;-75.545554,6.340448?annotations=distance,duration

Debido a que es una API opensource tiene algunas limitaciones en la cantidad de peticiones entonces el uso es exclusiva para entornos de prueba.

Documentación API OpenStreetMap : https://wiki.openstreetmap.org/wiki/API\_v0.6

Datos adicionales :

OSMnx : Esta es una libreria de python que a futuro puede ser util para la visualizacion de datos, modelar, trazar y analizar redes de calles. OSMnx se basa en GeoPandas, NetworkX y matplotlib e interactúa con las API de OpenStreetMap. Se ve de importancia para tener en cuenta a futuro

Documentacion: https://osmnx.readthedocs.io/en/stable/

BLog :https://geoinnova.org/blog-territorio/analisis-redes-de-transporte-phyton-osmnx-networkx/

MARTES 01 febrero

https://router.project-osrm.org/route/v1/car/-75.538971,6.345817;-75.541670,6.343082?steps=true&annotations=distance,duration

OSRm ofrece cordenadas por tramos donde hay intersecciones pero no son continuas con las distancias.

Ejemplo de tramos:

6.34582, -75.538975

6.345577, -75.539199

6.345493, -75.539276

6.345096, -75.539565

6.344239, -75.539588

6.343657, -75.541735

6.343084,-75.541664

Graphhoper tiene un limite de peticiones en el servicio gratuito, se ve una buena opcion(por revisar)

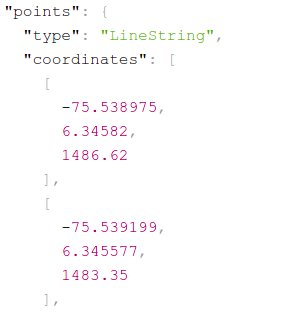
https://graphhopper.com/maps/

YOURS al parecer la api no funciona(no carga)

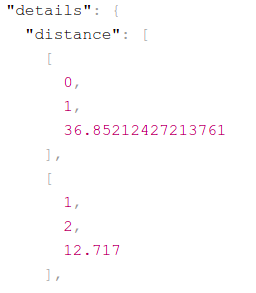
SABADO 5 de febrero.

El codigo se completo usando la API de Graphhopper, esta es una API de paga con un servicio gratuito. Este servicio nos ofrece 500 peticiones por dia, por cada peticion solo se puede hacer consultar 5 nodos, en caso de necesitar mas nodos se deben hacer varias peticiones.

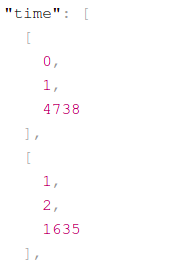
Graphhopper nos ofrece una buena cantidad de información como distancias, tiempo aproximado basado en la velocidad maxima que se es permitida por tramos, nos ofrece coordenadas de cada tramo, la altitud de cada punto y instrucciones por tramo para la navegación.

Coordinates : [[longitude, latitude, elevation]]

En los detalles del camino se puede denotar la distancia por tramos e intervalos que coinciden con las coodernadas y el tiempo.

El 3er dato es la distancia en metros.

Tambien se obtiene el tiempo:

El 3er datos es el tiempo en milisengundos.

EJEMPLO de request:

<https://graphhopper.com/api/1/route?point=6.345817,-75.538971&point=6.340448,-75.545554&point=6.341181,-75.543221&&elevation=true&points_encoded=false&details=time&details=distance&calc_points=true&key=api_key>

Para obtener el api\_key = debe logguearse y alli le daran una llave para hacer consultas.

DOCUMENTACION DE GRAPHHOPPER : https://docs.graphhopper.com/#operation/getRoute