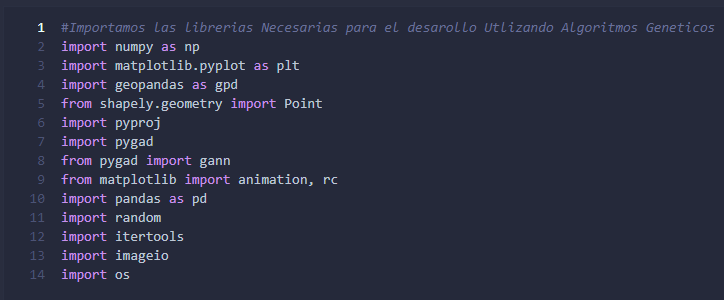
Solución al problema del Viajero.

Para la solución de este problema atreves de algoritmos Genéticos y Colonias de Hormigas primero debemos preparar todos los datos e información necesaria para solucionar el problema.

Empezamos como siempre cargado nuestras librerías necesarias

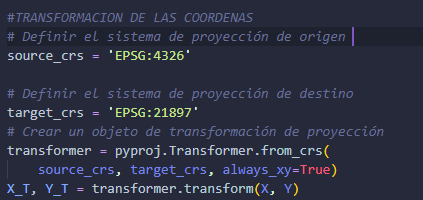


Ahora obtenemos los datos de las coordenadas de las ciudades solicitadas, esto se hizo con la página web <https://www.geodatos.net/coordenadas/colombia> de donde se extrajeron todos los datos necesarios

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

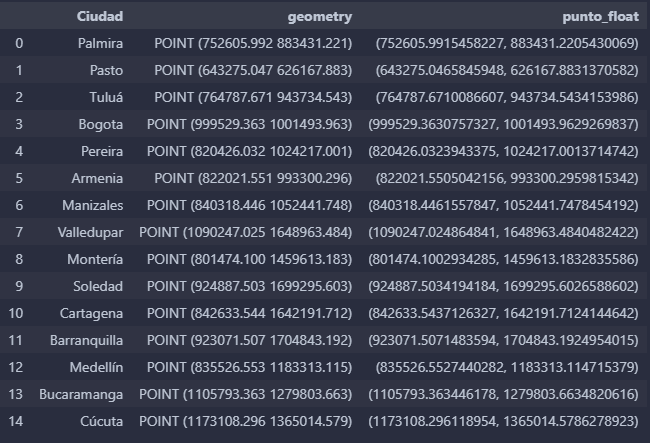
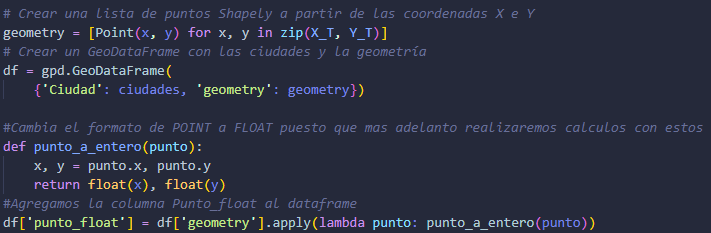
Estas coordenadas fueron transformadas para que coincidieran con la escala de dimensión de archivo shapefile de Colombia el cual tenia un formato CRS de tipo EPSG:21897 y las coordenadas que normalmente encontramos están en formato EPSG:4326, usando la librería pyproj que es una librería que nos permite hacer transformación para coordenadas



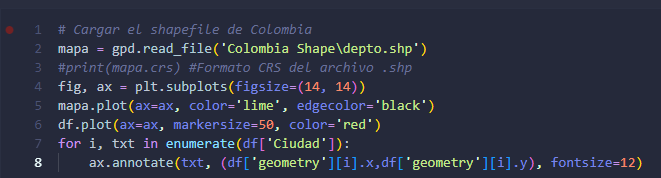
De esta forma obtenemos unas coordenadas proyectadas al formato EPSG:21897, un ejemplo de algunas es:



Finalmente, estas coordenadas, las ciudades y un casteo de las coordenadas a formato float (Esto se hace debido a que las coordenadas obtenidas por obtenidas tienen el formato POINT de la librería shapely y no son de un formato apto para realizar operaciones numericas)



Con esto y un Archivo que contenga el contorno y la división política de Colombia podemos crear un mapa de Colombia con las ciudades ubicadas en el mapa



Obtenemos una salida así:

Mapa

Descripción generada automáticamente

Para la solución de este ejercicio tómanos como medio transporte un Chevrolet Camaro SS el cual tiene las siguientes características relacionadas al problema:

Su rendimiento en KM por galón es: 29.15Km/galón.

Nuestro automóvil en Colombia paga la tarifa de categoría 1. Este costo será de Aproximadamente $8000 COP (Esto puede variar mucho o poco entre peajes, pero se fijó en ese valor para simplicidad de los cálculos)

El precio de la gasolina a la fecha de realización de este trabajo (marzo 15 del 2023) es de:

$9.664 pesos colombianos por galón.

Este valor se obtuvo de: <https://www.elcolombiano.com/negocios/valor-del-galon-de-gasolina-en-medellin-y-otras-ciudades-de-colombia-con-alza-de-200-pesos-en-octubre-KI18762697>

EL sueldo por Hora de nuestro chofer a la fecha de realización de este trabajo (marzo 15 del 2023) será de:

$1368,39 COP por Hora

Este valor se obtuvo de la pagina web <https://co.indeed.com/cmp/Redetrans-S.a./salaries/Conductor-a#:~:text=El%20sueldo%20promedio%20que%20un,en%20los%20últimos%2036%20meses>.

Ahora se calcula la matriz de costos totales, esta matriz contiene el valor de viajar entre cada ciudad a la otra y es calculada de otras 3 matrices:

MATRIZ DE DISTANCIAS ENTRE LAS CIUDADES:MDC

Esta matriz se calcula con la función de Haversine esta función nos permite calcular la distancia entre 2 ciudades en kilómetros para nuestro contexto, hay que tener en cuenta que esta función calcula la distancia que es básicamente una line recta entre las ciudades y no el trayecto entre las ciudades el cual no es línea recta y puede ser un poco mas grande que la distancia.

Texto

Descripción generada automáticamente

Asi se obtiene la matriz de distancias MDC

Texto

Descripción generada automáticamente

Las siguientes 2 matrices se hicieron a mano en Excel consultando los datos de la página https://www.peajesencolombia.com

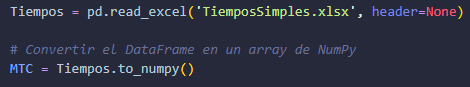
MATRIZ DE # DE PEJAES ENTRE LAS CIUDADES:MNP

Usando la página mencionada atrás se extrajeron los datos del número de peajes entre las ciudades, Luego se cargó el Excel y se especifica que no use el parámetro header para que no use la primera fila como nombres de las columnas   
Texto

Descripción generada automáticamente

MATRIZ DE TIEMPO ESTIMADO ENTRE CIUDADES:MTC

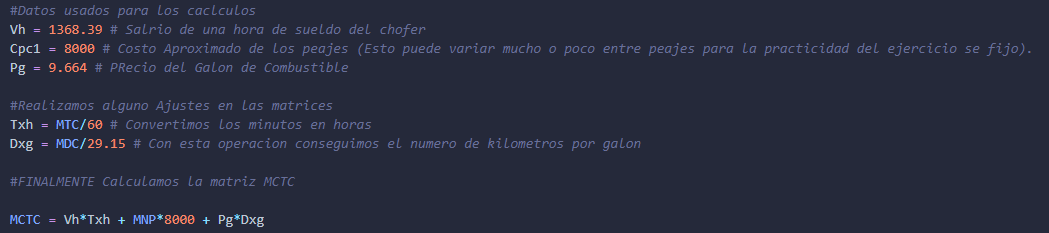
De forma similar a la anterior matriz y con la misma pagina web y cargando los datos de la misma manera obtenemos la matriz de tiempos en los cuales se guardaron los valores en Minutos de trayecto entre ciudades



Vamos a hablar un poco sobre las matrices, la matriz de distancia MDC contiene las distancias en Km y se usa para calcular el dinero requerido para comprar la gasolina, para esto multiplicamos el valor del galón de la gasolina por la Matriz divida por 29.15 km que es el rendimiento en galones de nuestro vehículo de transporte.

Luego la matriz de Peajes por ciudad se multiplica por el costo de los peajes

Finalmente, la matriz de tiempos entre ciudades se divide por 60 para que los valores queden en horas y se multiplica por el valor de hora de trabajo del chofer.



Y de esta forma obtenemos la matriz de costos Totales de viaje entre ciudades.

Con todo esto hecho ya podemos aplicar algoritmos genéticos y colonias de hormigas para solucionar el problema del viajero Colombian versión.

Empecemos con Algoritmos genéticos antes de mostrar la instancia GA de algoritmos genéticos hablemos de la función de fitness o función de minimizado.

Para empezar esta función tenia la tarea de calcular el costo de la ruta encontrada por el algoritmo genético por cada iteración del código evaluando la solución en la matriz de costos y también se añadieron algunas restricciones:

1. La ruta debe contener todas las ciudades

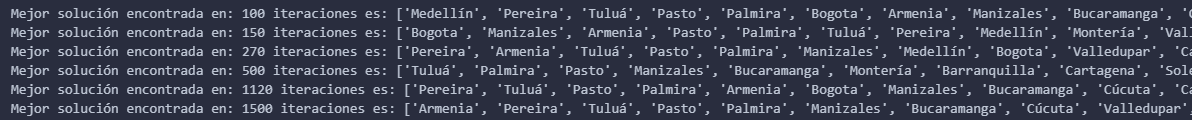
2. La ruta solo puede pasar una única vez por ciudad es decir no se permiten ciudades repetidas la ruta.

Texto

Descripción generada automáticamente

Luego se implementa el algoritmo genético que esta sujeto a un numero de Generaciones para poder guardar y comparar como cambia según el número de iteraciones y se ejecuta



Así obtenemos soluciones de esta forma  


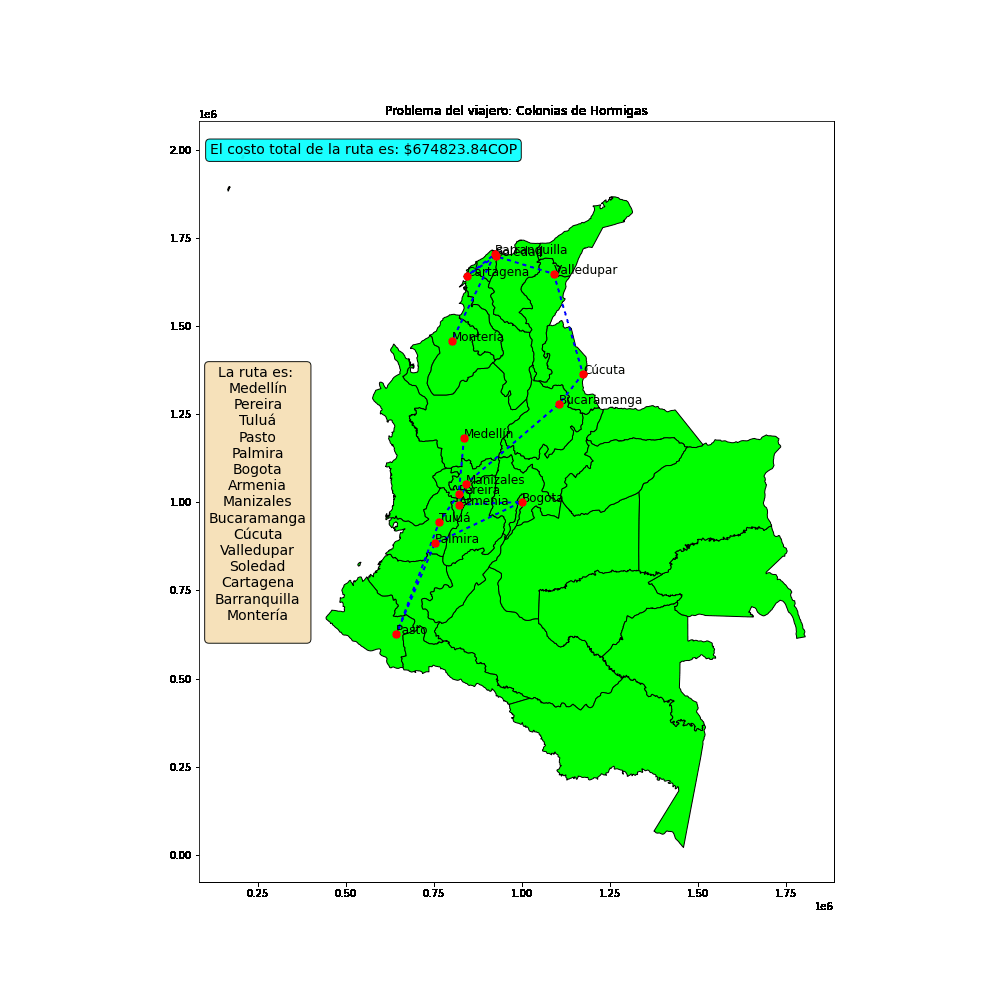
Calculamos el costo de cada ruta  
Texto

Descripción generada automáticamente

Un conjunto de letras blancas en un fondo blanco

Descripción generada automáticamente con confianza media

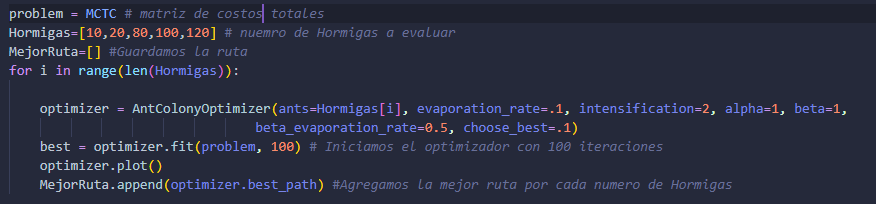
Finalmente usando las funciones de la librería matplotlib y la librería Imageio Guardamos imágenes de las rutas conseguidas y las unimos todas en un gif



Ahora veamos solución por colonias de hormigas

La implementación este problema en colonias de hormigas es más sencillo ya que existe una clase llamada AntColony que resuelve problemas por colonias de hormigas y solo es necesario instanciar el optimizador y dar los valores a la función.

De forma similar a la anterior ahora usamos varios números de hormigas para ver cuanto cambian las rutas de una a otra



Luego se grafica la convergencia a una solución del algoritmo

Gráfico

Descripción generada automáticamente

Aquí la solución convergió en la iteración 60

Luego calculamos el costo del recorrido

Texto

Descripción generada automáticamente

Finalmente usando las funciones de la librería matplotlib y la librería Imageio Guardamos imágenes de las rutas conseguidas y las unimos todas en un gif

