1. **Introducción**

El comercio electrónico ha experimentado una transformación significativa en los últimos cinco años, consolidándose como un canal fundamental para las empresas a nivel global. Entre los últimos años, con la evolución tecnológica, los cambios en los hábitos de consumo y la aceleración provocada por la pandemia de COVID-19 han sido factores determinantes. Las empresas han adoptado soluciones digitales avanzadas, como inteligencia artificial, análisis de datos y plataformas multicanal, para mejorar la experiencia del cliente. Además, se ha incrementado la competencia en el sector, con la entrada de nuevos actores y la diversificación de modelos de negocio, como el comercio social y las suscripciones personalizadas.

En este periodo, los consumidores han mostrado una creciente preferencia por las compras en línea debido a la comodidad, la personalización y la disponibilidad de opciones. Este cambio ha impulsado a las empresas a desarrollar estrategias de marketing digital más sofisticadas, integrando SEO, redes sociales y campañas automatizadas. Paralelamente, la importancia de la logística y las entregas rápidas se ha intensificado, fomentando la colaboración con terceros. Este contexto ha consolidado el comercio electrónico como un pilar estratégico para las empresas, tanto para capturar nuevos mercados como para adaptarse a las expectativas cambiantes de los consumidores.

La empresa IA DELIVERY S.L. con sede en Madrid, se dedica a la manufactura de productos alimenticios perecederos de corta duración. Han identificado que una parte significativa de sus costes operativos proviene de la externalización de servicio de distribución a terceros. Tras realizar un análisis exhaustivo de estos costes, han decidido internalizar este proceso y crear su propia flota de vehículos, optimizando así la distribución de sus productos desde sus instalaciones centrales.

Con el fin de reducir costos y mejorar la eficiencia, la empresa busca desarrollar una herramienta de optimización de rutas. Esta herramienta deberá calcular las rutas más eficientes para la distribución de sus productos, teniendo en cuenta factores como la cantidad mínima de vehículos necesaria, las distancias a recorrer, el tiempo de entrega y la capacidad de los vehículos. El objetivo final es minimizar el uso de recursos y garantizar una distribución efectiva con la menor inversión de flota propia.

1. **Descripción general**
   1. **Objetivos**
      1. **Objetivo general**

Se busca minimizar los costes totales generados por el reparto de artículos vendidos y maximizar el número de entregas con la flota de vehículos disponibles, con el objetivo de alcanzar el máximo beneficio disponible.

* + 1. **Objetivos específicos**
* Realizar el análisis del histórico de datos de entregas y clientes de la empresa para conocer el contexto de ventas de la empresa.
* Desarrollar un modelo que planifique las rutas de entrega, minimizando el coste total y maximizando el número de entregas realizadas cumpliendo las restricciones de capacidad, autonomía y vehículos disponibles.
* Suponiendo que la empresa tenga que llevar acabo una reducción de su flota de vehículos disponibles para la entrega de pedidos (por causas de economía o mantenimiento), optimizar las rutas para mantener el máximo de los beneficios.
* Desarrollar un modelo de demanda de pedidos que predice el número de pedidos esperados para el próximo mes y desarrollar las rutas para cumplir con la demanda prevista y ajustarse a los pedidos futuros.
  1. **Business Criteria**

Para conocer si con el paso del tiempo el modelo está funcionando correctamente, se podrá comparar los datos de valores de tiempo y distancias entre las rutas, teniendo en cuenta que se busca el mínimo de tiempo invertido para la entrega de pedidos, y el máximo de entregas realizadas.

1. **Data understanding**

Proporciona la base para tomar decisiones informadas sobre cómo procesar, transformar y analizar los datos en las etapas posteriores del proyecto. Una comprensión sólida de los datos asegura un análisis más efectivo y resultados de mayor calidad. Para el proyecto nos ceñiremos a los datos proporcionados únicamente.

* 1. **Data exploration report**

Para el desarrollo del proyecto, disponemos de 6 archivos .csv con el contenido de datos a estudiar para la resolución y obtención del objetivo.

* **df\_distance\_km:** contiene los datos entre las localizaciones en km

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Tipo** | **Descripción** |
| cliente\_k | Numérico | Contiene la distancia de Cliente\_k a Cliente\_X |
| almacen | Numérico | Contiene la distancia de Almacen a Cliente\_X |

* **df\_distance\_min:** contiene las datos de tiempo entre las localizaciones en minutos

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Tipo** | **Descripción** |
| cliente\_k | Numérico | Contiene el tiempo de Cliente\_k a Cliente\_X |
| almacen | Numérico | Contiene el tiempo de Almacen a Cliente\_X |

* **df\_historic\_order\_demand:** registro del histórico de pedidos realizados

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Tipo** | **Descripción** |
| cliente | Categórica | Contiene el cliente que realiza el pedido |
| mes\_anio | Fecha | Contiene el dato de mes y año en que se realiza el pedido |
| order\_demand | Numérico | Contiene el peso en kg del pedido |

* **df\_location:** contiene los datos de localizaciones

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Tipo** | **Descripción** |
| cliente | Categórica | Contiene el cliente que realiza el pedido |
| latitud | Numérico | Contiene el valor de la latitud del cliente |
| longitud | Numérico | Contiene el valor de la longitud del cliente |

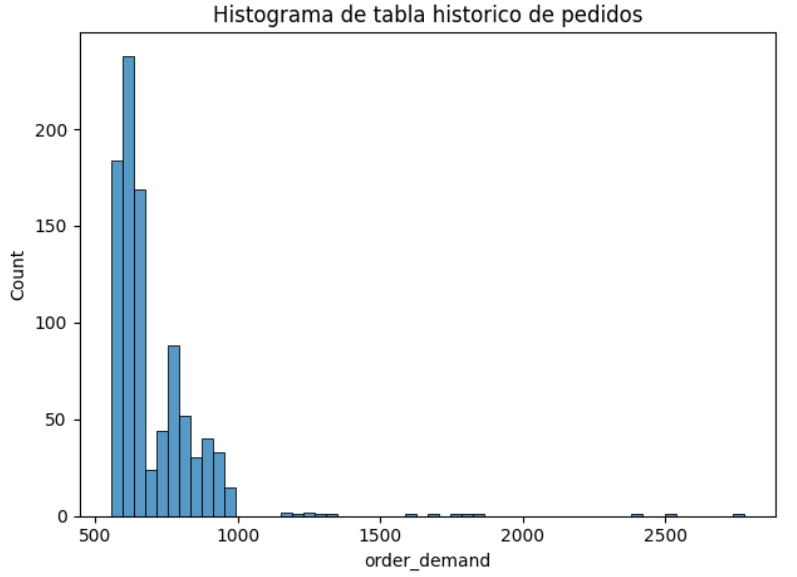
* **df\_orders:** registro del mes de diciembre de pedidos realizados

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Tipo** | **Descripción** |
| cliente | Categórica | Contiene el cliente que realiza el pedido |
| mes\_anio | Fecha | Contiene el dato de mes y año en que se realiza el pedido |
| order\_demand | Numérico | Contiene el peso en kg del pedido |

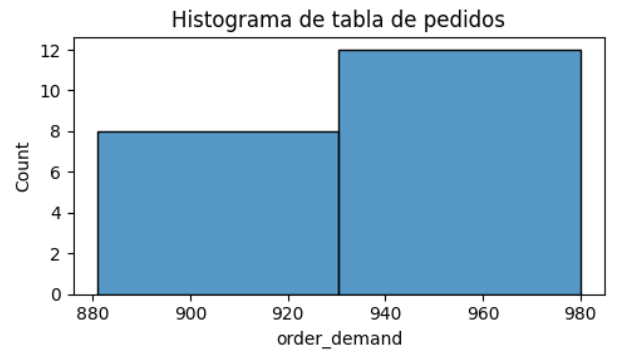
* **df\_vehicle:** contiene los datos de los vehículos disponibles

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Tipo** | **Descripción** |
| vehiculo\_id | Numérica | Contiene el valor del id único del vehículo |
| capacidad\_kg | Numérica | Contiene el valor de la capacidad máxima que permite |
| costo\_km | Numérico | Contiene el valor de coste por km de cada vehículo |
| autonomia\_km | Numérico | Contiene el valor de la autonomía disponible de cada vehículo |

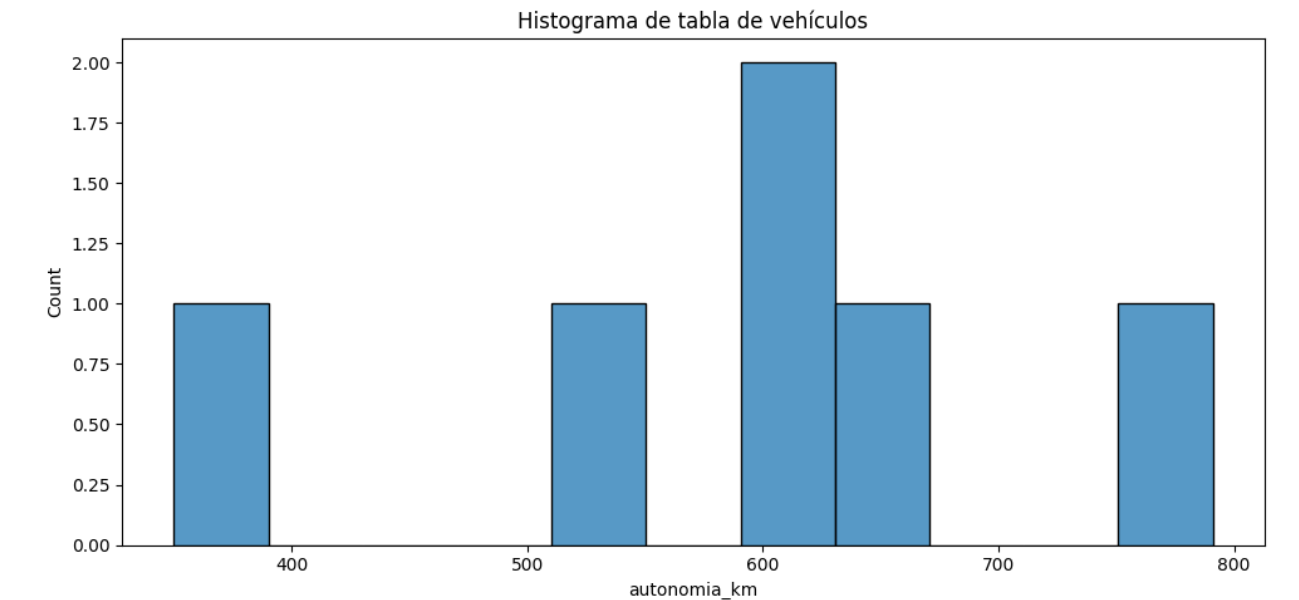
* 1. **Infographics**
* **Histograma de tablas historico de pedidos:** Encontramos la comparación entre el peso y la cantidad del pedido de los clientes, se observa que la mayoria de pedidos, tienen un peso entre 500 y 1000 kg.



* **Historico de tabla de pedidos:** Encontramos la comparación entre el peso y la cantidad de los pedidos realizados por los clientes en el último mes y se observa que la mayoria de pedidos, tienen un peso cercano a los 1000 kg.

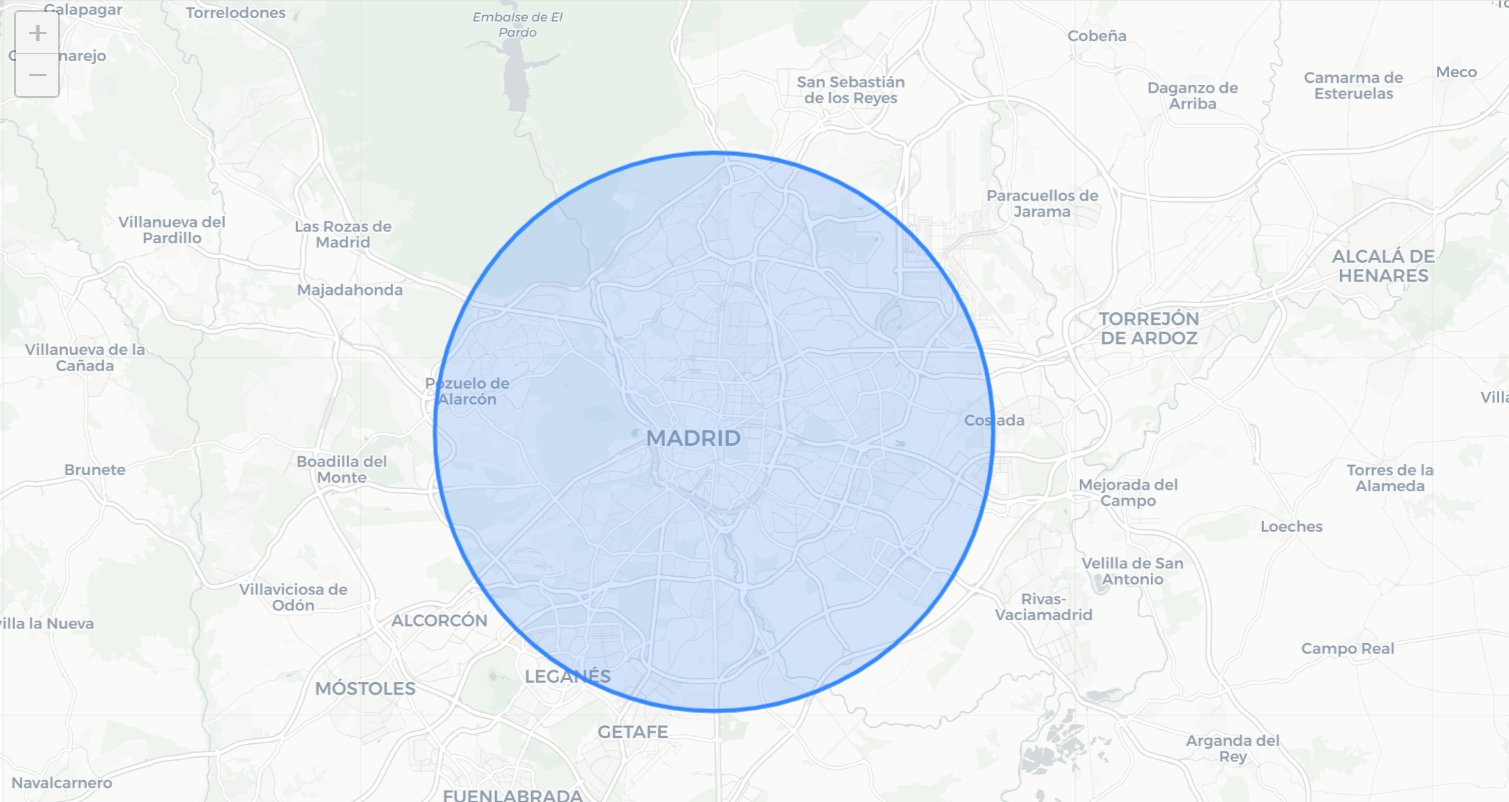


* **Histograma de tabla de vehículos:** Se muestra la autonomia de los diferentes vehiculos.

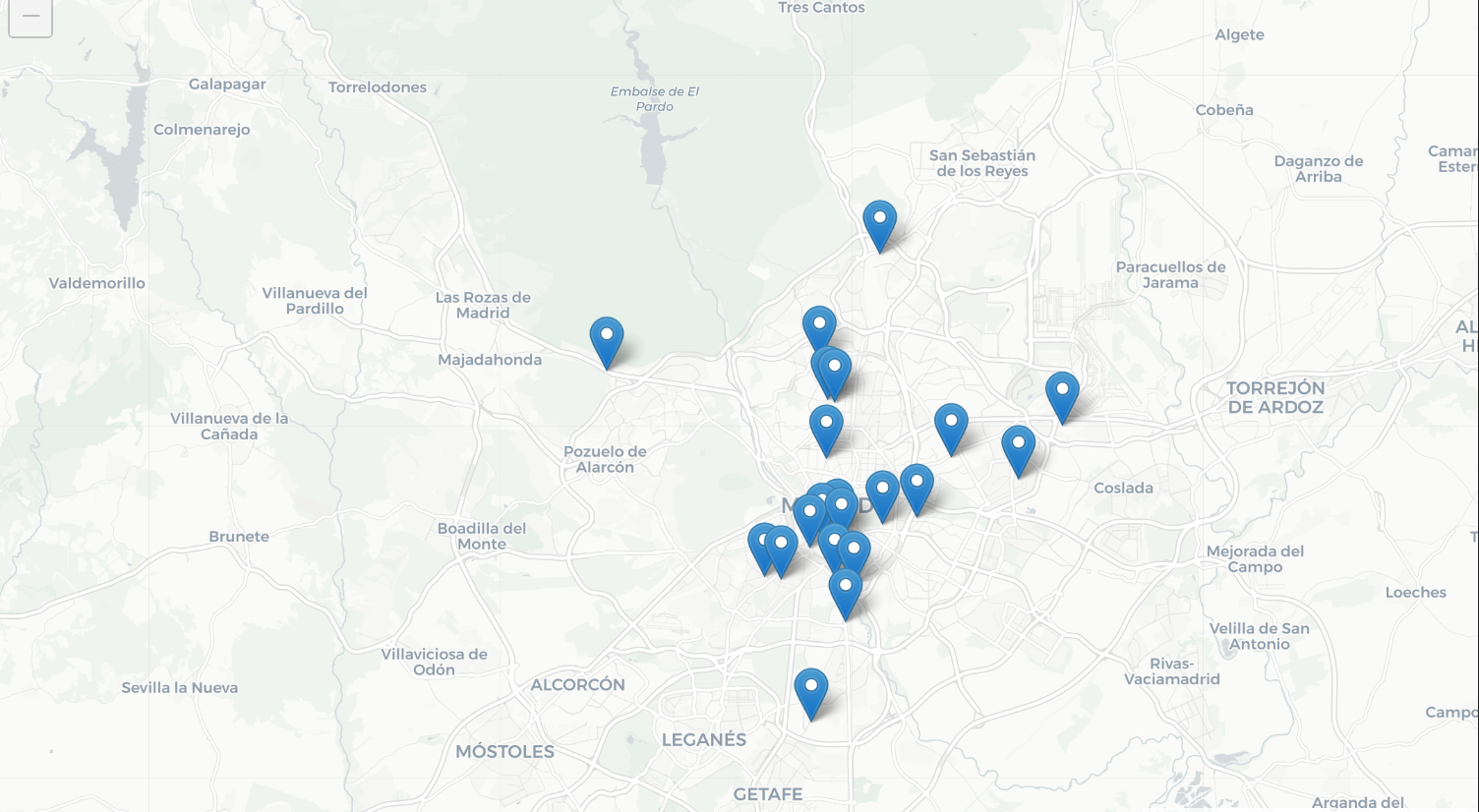


* **Mapa de ubicación de clientes:** Se muestran los distintas ubicaciones de los clientes en un mapa, primero se ve el radio de donde se encuentran las ubicaciones y por último los puntos exactos de las locaclizaciones.

Radio de ubicaciones



Radio de locaclizaciones



1. **Data Preparation**

**4.1 Clean Data and Construct Data**

La preparación de datos consiste en la limpieza y manipulación de los datos en crudo para mejorar el procesamiento y el análisis de dichos datos.

Para el caso número 3 del proyecto (predicción de demanda de clientes de enero de 2025) se ha tenido que realizar un análisis exploratorio del archivo **df\_historic\_order\_demand** para comprobar que los datos estaban correctamente.

Se ha encontrado que la columna order\_demand tenía valores nulos por lo que se ha procedido a su eliminación.

* 1. **Format Data**

Para el caso número 3 del proyecto (predicción de demanda de clientes de enero de 2025) se aplica un algoritmo de regresión (random forest regressor) para poder predecir la demanda y dicho algoritmo no acepta valores categóricos, por lo que las columnas mes\_anio y clientes han tenido que ser transformadas a numéricas.

Para ello, en el caso de la columna clientes se ha utilizado la función get\_dummies, que coge todos los valores únicos de una columna y los convierte en nuevas columnas cuyo nuevo valor será 1 o 0 como se muestra en la siguiente imagen:



Para el caso de la columna mes\_anio, cuando se hizo el análisis exploratorio se vio que dicha columna estaba compuesta por mes y año en este formato “MM-YYYY” y todos los valores de dicha columna eran únicos por lo que aplicar get\_dummies no era la mejor solución.

Se optó por separar mes y año en dos columnas distintas y así poder convertirlas a numéricas.

1. **Modeling**

**A. Clean Data and Construct Data**

La exploración de los modelos se lleva a cabo con el indicado RMSE (Error medio), en el caso de los algoritmos Djikstra, PuLP y genético no se mide el RMSE ya que no es un algoritmo de predicción sino de optimización por lo que el resultado será la ruta más rápida entre una serie de puntos dados.

**B. Build Model**

**1) Algoritmo Dijkstra**

Para el modelo 0, se va a aplicar el algoritmo de optimización Djikstra que es un método utilizado para encontrar el camino más corto en un grafo ponderado (las aristas tienen un peso, que puede representar costo, distancia o cualquier métrica relevante) con pesos no negativos.

Funciona de la siguiente manera:

* El algoritmo de Dijkstra básicamente inicia en el nodo que escojas (el nodo de origen) y analiza el grafo para encontrar el camino más corto entre ese nodo y todos los otros nodos en el grafo.
* El algoritmo mantiene un registro de la distancia conocida más corta desde el nodo de origen hasta cada nodo y actualiza el valor si encuentra un camino más corto.
* Una vez que el algoritmo ha encontrado el camino más corto entre el nodo de origen y otro nodo, ese nodo se marca como "visitado" y se agrega al camino.
* El proceso continúa hasta que todos los nodos en el grafo han sido añadidos al camino. De esta forma, tenemos un camino que conecta al nodo de origen con todos los otros nodos siguiendo el camino más corto posible para llegar a cada uno de ellos.

**2) Algoritmo de Programación Lineal**

Para el modelo 1, se va a aplicar la librería PuLP de Python, que encuentra soluciones óptimas al problema como restricciones lineales.

PuLP permite indicar el tipo de problema que hay que optimizar mediante palabras reservadas de la propia librería, maximización (LpMaximize) o minimización (LPMinimize), que deberán usarse cuando comenzamos a definirlo. Acto seguido se definirá la función objetivo y sus restricciones para que se pueda iniciar la búsqueda de posibles soluciones.

**C. Access Model**

**1) Algorimo Djikstra**

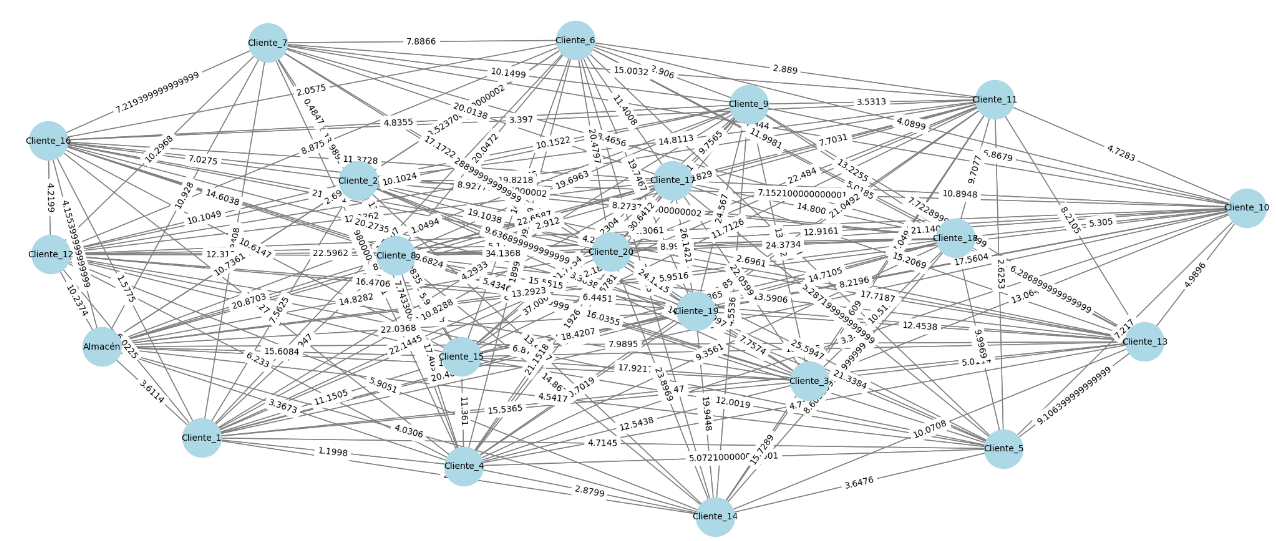
Partimos de los datos de distancias entre los distintos clientes (medida en kilómetros) en forma de matriz.



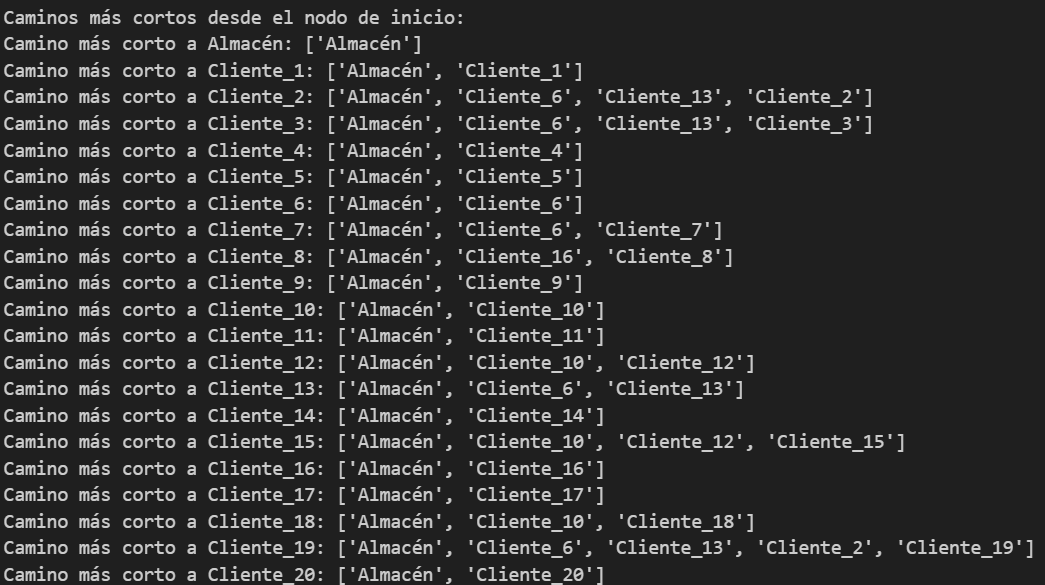
En la imagen se puede observar que la diagonal principal son todos ceros ya que la distancia entre Cliente\_1 y el mismo es cero.

Por otro lado, se aprecia que hay más ceros aparte de los de la diagonal ya que es posible que no haya una ruta directa entre dos clientes, es decir, que no hay camino.

Para hacernos una idea de cómo están relacionados todos los clientes, creamos un diagrama que muestra la conexión entre todos los nodos y sus correspondientes distancias como se muestra en la siguiente imagen.



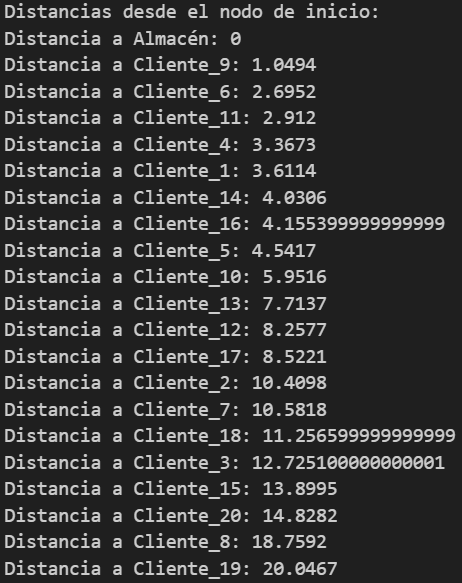
* Nuestro primer caso de uso consiste en calcular la ruta más corta (km) entre el almacén y los distintos clientes. Estos han sido los resultados:



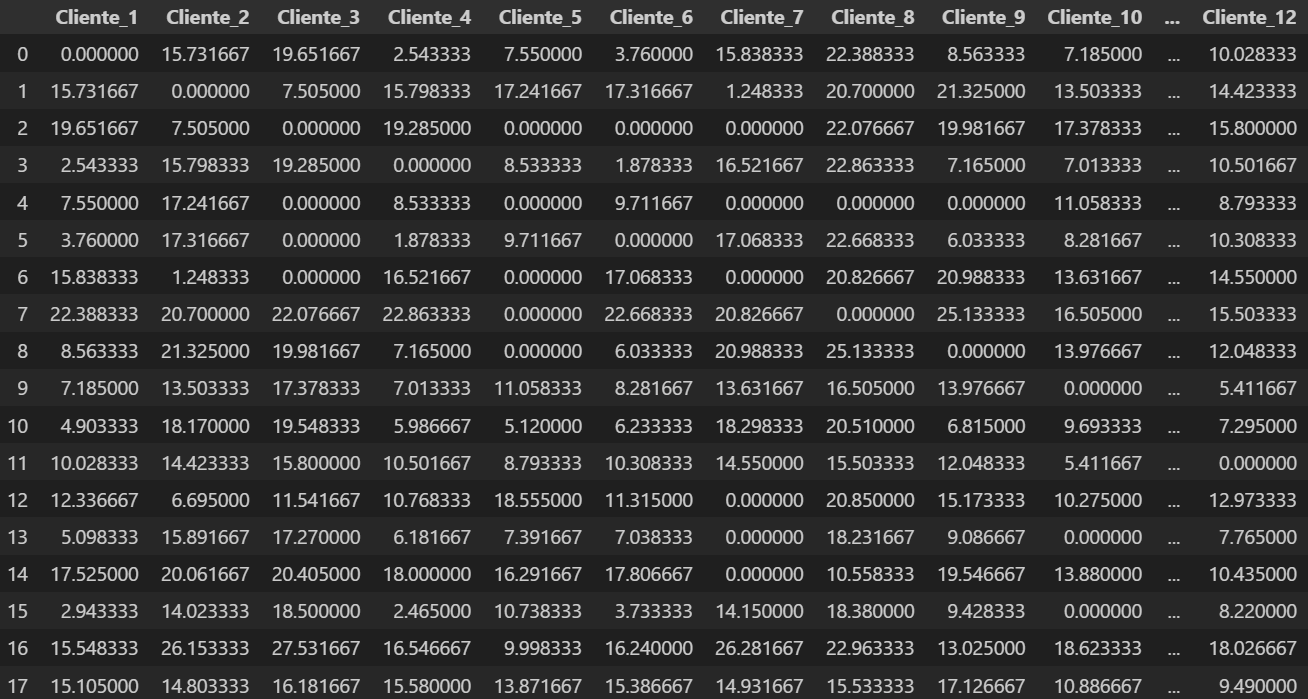
Se observa que hay casos en los que la ruta más corta es directamente ruta que hay entre ellos mismos y en otros casos pasa por otros clientes hasta llegar al nodo final.

Por ejemplo, en el camino más corto para llegar al Cliente\_2 la distancia entre dicho cliente y el almacén es de **10.7361 km,** pero Djikstra ha encontrado una ruta más rápida, pero pasando antes por el Cliente\_6 y el Cliente\_13 y la distancia total recorrida por este camino es de **10.4098 km,** que es ligeramente menor que la primera distancia.

En la siguiente imagen se muestra una lista de distancias entre ambos puntos:



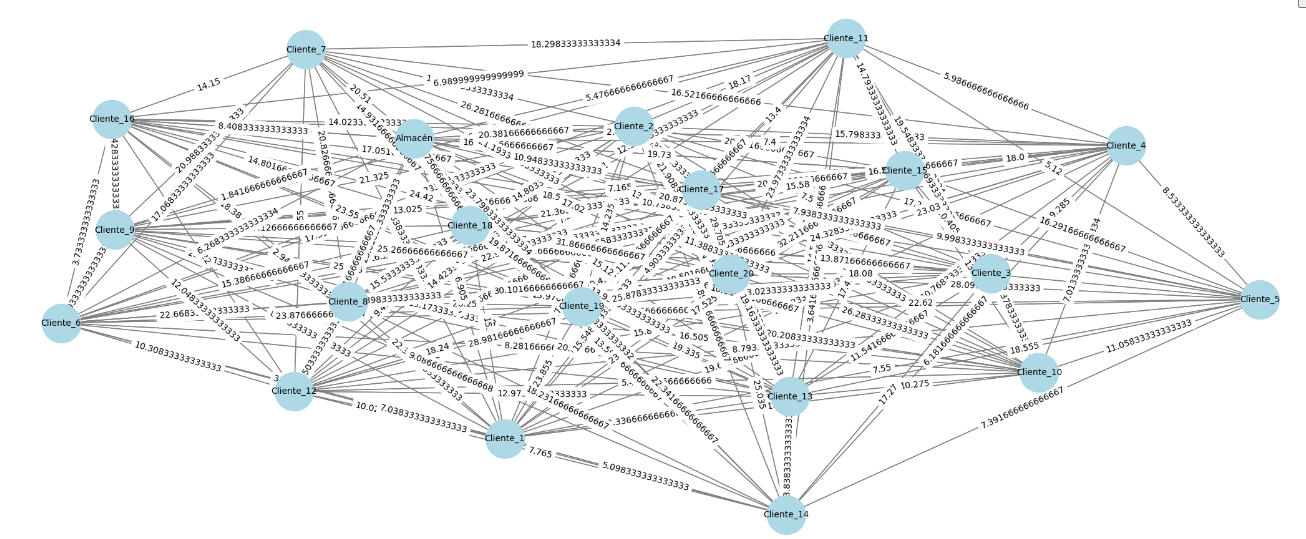
* Nuestro segundo caso de uso consiste en calcular la ruta más rápida en término de tiempo entre el almacén y los distintos clientes. Partimos de estos datos:



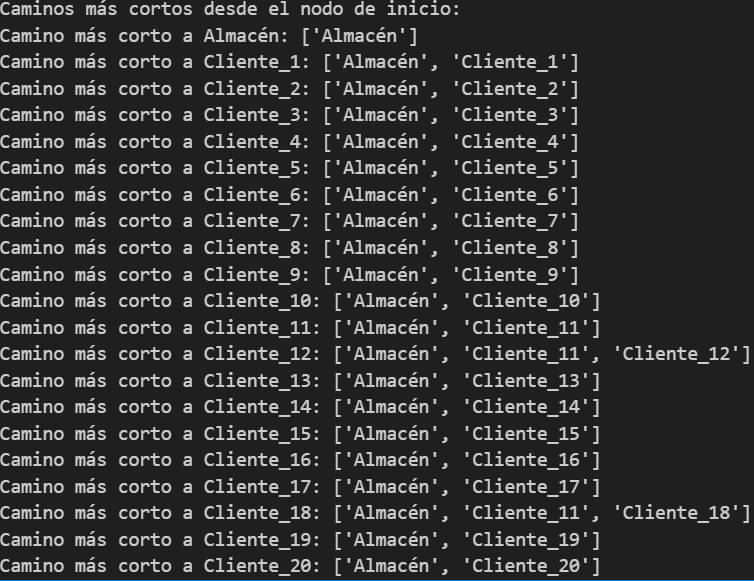
En la imagen se puede observar que la diagonal principal son todos ceros ya que la distancia en tiempo entre Cliente\_1 y el mismo es cero.

Por otro lado, se aprecia que hay más ceros aparte de los de la diagonal ya que es posible que no haya una ruta directa entre dos clientes por lo que el tiempo es 0.

Para hacernos una idea de cómo están relacionados todos los clientes, creamos un diagrama que muestra la conexión entre todos los nodos y sus correspondientes distancias en minutos como se muestra en la siguiente imagen.



Aplicado el algoritmo se han obtenido los siguientes resultados:

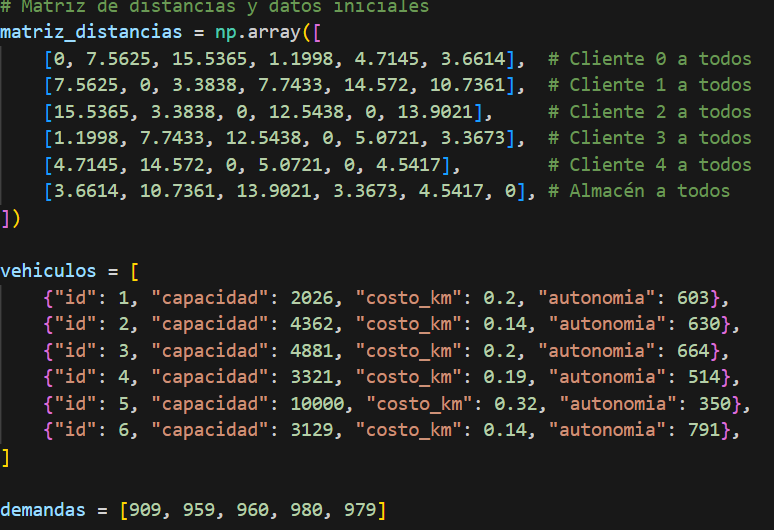


Al contrario que en el primer caso de uso, nos encontramos con que en general la ruta más rápida en término de tiempo es la ruta entre ambos puntos sin intermediarios.

Se observa que los resultados no son los mismo que en el anterior caso de uso ya que una distancia en kilómetros más corta no significa que el tiempo sea menor ya que puede haber obstáculos como semáforos o muchas curvas etc.

**2) Algoritmo de Programación Lineal**

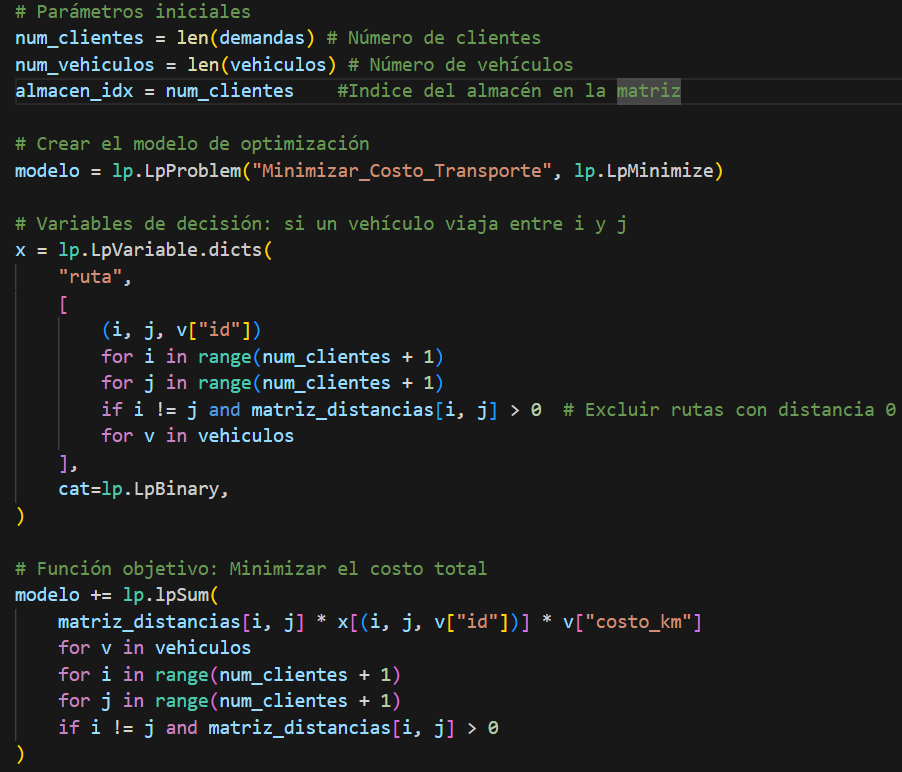
Partimos de los siguientes datos de prueba:



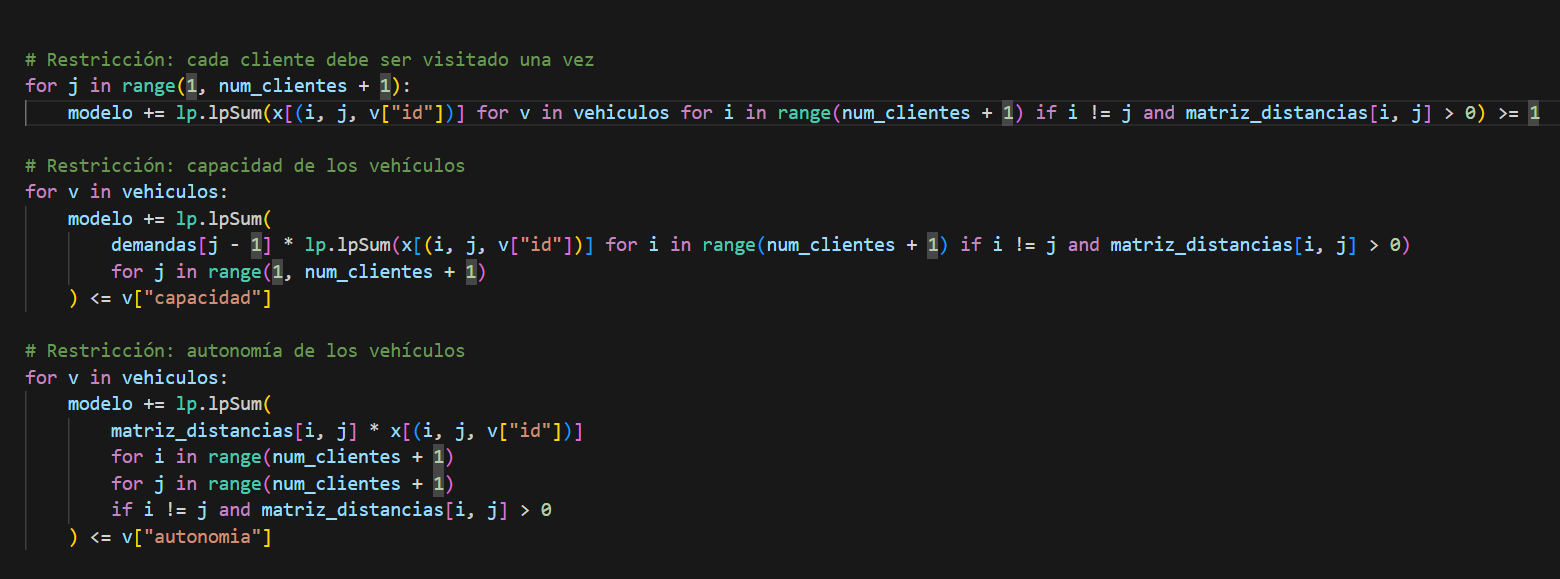
* La matriz de distancias representa la distancia en kilómetros entre clientes y el almacén. Los registros que son ceros significan que no hay ruta directa entre dichos puntos
* Los vehículos contienen información como su capacidad máxima en kilogramos, su autonomía en kilómetros y el costo por kilómetro.
* Las demandas contienen la demanda de pedido de cada cliente en kilogramos.

Nuestro caso de uso consiste en calcular las posibles rutas de cada vehículo que pasen por el máximo número de clientes y con el menor costo posible.

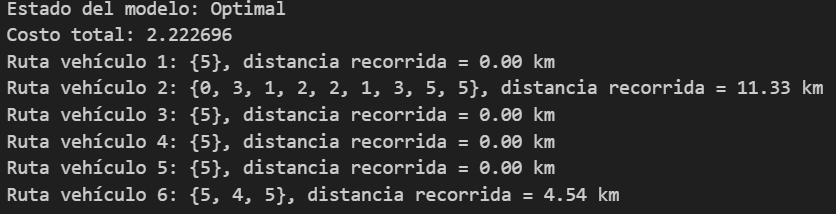
Para ello definimos nuestra función objetivo que en este caso es la de minimizar costes y creamos nuestras variables decisión que serán si un vehículo viaja entre dos clientes, como se muestra en la imagen:



Acto seguido procedemos a crear las restricciones del modelo que serán las relativas a capacidad máxima y autonomía de cada vehículo y que cada cliente debe ser visitado como mínimo una vez.



Ahora que tenemos nuestro objetivo y restricciones definidos pasaremos a resolver nuestro problema y estas han sido las rutas generadas:



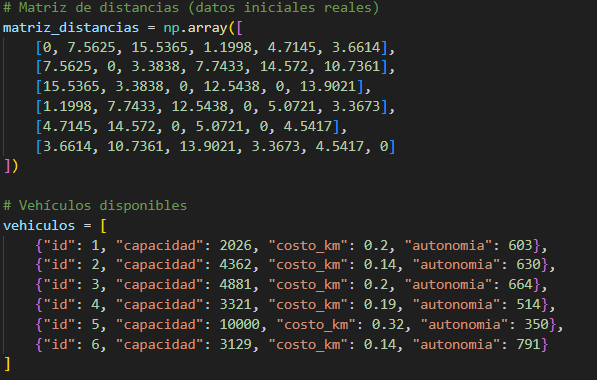
Como se puede observar, el estado del modelo es óptimo por lo que nuestras restricciones y objetivo han funcionado.

Hay cuatro vehículos que no tienen rutas asignadas y el vehículo dos es el que más clientes recorre. También se observa que el vehículo dos no pasa por el cliente 4 (que en realidad es el 5) y el vehículo 6 solo pasa por dicho cliente por lo que nuestra restricción funciona.

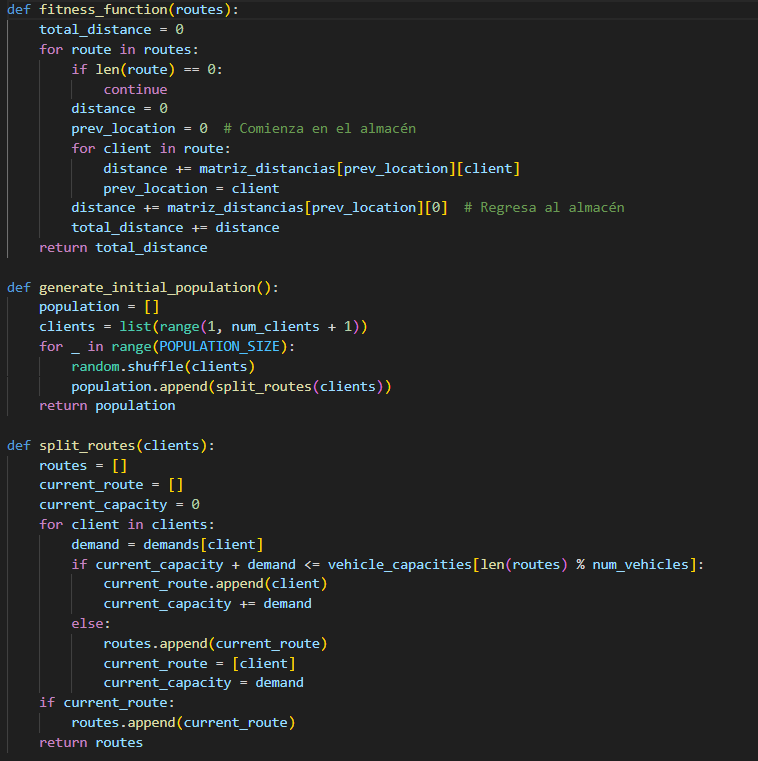
**3) Algoritmo Genético**

Partimos de los mismos datos de prueba del anterior algoritmo, para la realización del algoritmo genético.

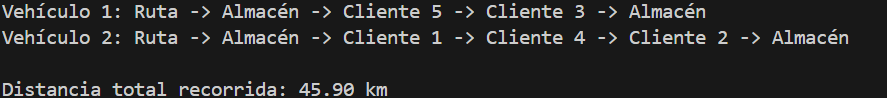
- Nuestro caso de uso es un método de optimización y búsqueda de las mejores rutas, que utiliza procesos como la selección, cruzamiento y mutación para encontrar las rutas óptimas de los vehículos para que repartan a más clientes con el menor coste.



Definimos una función de aptitud que minimiza la distancia total recorrida por los vehículos, asignando clientes a rutas según la capacidad de cada uno. Las soluciones se generan aleatoriamente en clientes y las variables de decisión son las rutas asignadas a los vehículos.



-Aquí tenemos la solución de nuestro algoritmo y las rutas y la distancia generada:



**4) Algoritmo heurístico**

Mediante el uso de la biblioteca ORTools se utilizará como solución de ruteo con restricciones.

Primero realizamos la asignación de los valores a utilizar en el código, teniendo en cuenta una ligera modificación sobre los datos de entrada, de manera que quedará lo siguiente:

Tabla, Excel

Descripción generada automáticamente

Partiendo del archivo modificado y los Excel de df\_orders y df\_vehicle, generamos la definición de las variables a utilizar en la función create\_data\_model(), siendo data la variable diccionario de retorno que contiene los datos a utilizar.

Texto

Descripción generada automáticamente

A continuación se llama a la función main() que ejecutará la totalidad del código, estructurado de la siguiente manera:

1. Almacena la variable retorno de la función previamente comentada
2. Mapea nodos reales a índices internos para OR-Tools y representa las rutas posibles y restricciones
3. La función distance\_callback devuelve la distancia entre nodos usando data['distance\_matrix']. Si no existe una conexión directa, asigna una penalización alta
4. La función demand\_callback garantiza que los vehículos no excedan su capacidad
5. Añadir dimensión de autonomía para verificar que las rutas respeten la autonomía de los vehículos.

Texto

Descripción generada automáticamente

Finalmente la función print\_solution() muestra el resultado de las rutas generadas para que resulten interpretables y claras al usuario

Texto

Descripción generada automáticamente

Para la ejecución del código se hace una llamada a la función main

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza media

Actualmente aun existen errores en la solución generada que deben ser retocados y por tanto las rutas obtenidas no son optimas ni cumplen todos los requisitos.

Ejemplo de resultado:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

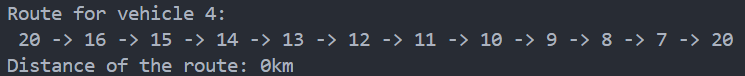
Interfaz de usuario gráfica, Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

Texto

Descripción generada automáticamente



Interfaz de usuario gráfica, Texto

Descripción generada automáticamente