

Transporte de Combustible

David Sánchez Iglesias
Manuel Alejandro Gamboa

June 2025

Introducción

Una empresa distribuidora de combustible necesita saber cuántos y qué tipo de vehículos necesita para cumplir con la demanda de sus clientes diariamente. Para ello cuenta con las rutas por las que transitarán sus futuros vehículos, y también de la demanda diaria de los mismos, así como la distancia a la que están de la empresa.

Las rutas están contenidas, en su mayoría por puentes, inclinaciones importantes y algunas no están pavimentadas, por lo que tienen restricciones en la cantidad de peso que admiten, así como el tipo de combustible que puede moverse por estas, ya que muchas rutas atraviesan zonas urbanas, y puede poner en peligro la vida de muchos ciudadanos, la altura máxima permitida para los vehículos también constituye una restricción, ya que estas rutas también pasan por debajo de ciertos puentes. Como cada cliente está en una ruta en específico, los vehículos deben cumplir con estas restricciones.

Por parte de los camiones, hay un límite de viajes que pueden hacer por día, esta cantidad está determinada por la cantidad de kilómetros que cada vehículo puede recorrer.

Para ayudar a esta empresa se puede tratar la situación, como un problema de satisfacción de restricciones, donde se tiene un conjunto de pedidos por parte de los clientes, y se quiere asignar para cada pedido, un vehículo capacitado para dicha labor, de manera que todos los pedidos tengan asignado un vehículo. Para darle solución se seguirá la siguiente dinámica:

1. Primero se recopilará información en la web en forma de documentos, ya que no se tiene sobre vehículos para transportar combustible. Para ello se utilizarán algoritmos de recuperación de información.
2. Luego se generarán vehículos con diferentes características, para seleccionar de los mismos aquellos que puedan satisfacer la demanda.
3. Dada una solución, que se le llamará solución inicial, haciendo una búsqueda local modificando los parámetros de la solución, y verificando que siga cumpliendo las restricciones, se buscará nuevas soluciones más convenientes, en este caso vehículos que cumplan y que sean más baratos. El

proyecto utilizado para afrontar el problema está alojado en la plataforma de GitHub en https://github.com/GamboaHMA/simulacion_ia_transporte

Recuperación de Información sobre Vehículos

Se recuperará una cantidad de documentos que no superen las 80000 palabras, luego en el siguiente paso se argumentará por qué. Se obtuvo como resultado los documentos que están contenidos en la carpeta **pdfdocuments**.

Generación de Vehículos

Para la generación de vehículos se implementará un agente, el cual recibe la información de los documentos que tratan fundamentalmente sobre vehículos para la transportación de combustible en forma de string como contexto, el límite para el plan gratuito es de 100000 palabras aproximadamente, por eso se limitó a que entre todos los documentos recuperados sumaran 80000 palabras, con un margen de 20000 palabras para agregarle otras cosas al contexto. Este agente usará la api de gemini-2.0-flash como modelo de lenguaje. Se escogió este modelo por su rapidez, generación de texto relevante, por su buen manejo del contexto, por la fácil integración y sencilla implementación para su uso.

Antes de pasar a generar vehículos se debe aclarar que se fijarán las Rutas, las cuales como se dijo al principio tienen una restricción de peso máximo, de altura y de tipo de combustible. También se fijarán las peticiones para la experimentación, ya que puede interpretarse como una demanda fija o diaria por parte de los clientes, que varía muy poco o nada.

Rutas:

```
Rute ID: 1, Altura: 3.8, PesoMaximo: 15000,
, Nodos: ['Node ID: 102, Rute ID: 1, Distance: 45', 'Node ID: 103, Rute ID: 1, Distance: 60', 'Node ID: 101, Rute ID: 1, Distance: 70']

Rute ID: 2, Altura: 3.8, PesoMaximo: 10000,
, Nodos: ['Node ID: 202, Rute ID: 2, Distance: 55', 'Node ID: 203, Rute ID: 2, Distance: 70', 'Node ID: 201, Rute ID: 2, Distance: 60']

Rute ID: 3, Altura: 3.8, PesoMaximo: 30000,
, Nodos: ['Node ID: 302, Rute ID: 3, Distance: 40', 'Node ID: 303, Rute ID: 3, Distance: 50', 'Node ID: 301, Rute ID: 3, Distance: 70']

Rute ID: 4, Altura: 3.4, PesoMaximo: 8000,
, Nodos: ['Node ID: 402, Rute ID: 4, Distance: 90', 'Node ID: 403, Rute ID: 4, Distance: 120', 'Node ID: 401, Rute ID: 4, Distance: 80']

Rute ID: 5, Altura: 4.0, PesoMaximo: 45000,
, Nodos: ['Node ID: 502, Rute ID: 5, Distance: 50', 'Node ID: 503, Rute ID: 5, Distance: 65', 'Node ID: 501, Rute ID: 5, Distance: 75']

Rute ID: 6, Altura: 3.7, PesoMaximo: 6000,
, Nodos: ['Node ID: 602, Rute ID: 6, Distance: 100', 'Node ID: 603, Rute ID: 6, Distance: 120', 'Node ID: 601, Rute ID: 6, Distance: 140']

Rute ID: 7, Altura: 3.85, PesoMaximo: 25000,
, Nodos: ['Node ID: 702, Rute ID: 7, Distance: 60', 'Node ID: 703, Rute ID: 7, Distance: 70', 'Node ID: 701, Rute ID: 7, Distance: 80']

Rute ID: 8, Altura: 3.65, PesoMaximo: 18000,
, Nodos: ['Node ID: 802, Rute ID: 8, Distance: 80', 'Node ID: 803, Rute ID: 8, Distance: 90', 'Node ID: 801, Rute ID: 8, Distance: 50']

Rute ID: 9, Altura: 3.9, PesoMaximo: 35000,
, Nodos: ['Node ID: 902, Rute ID: 9, Distance: 40', 'Node ID: 903, Rute ID: 9, Distance: 75', 'Node ID: 901, Rute ID: 9, Distance: 65']

Rute ID: 10, Altura: 3.55, PesoMaximo: 11000,
, Nodos: ['Node ID: 1002, Rute ID: 10, Distance: 80', 'Node ID: 1003, Rute ID: 10, Distance: 40', 'Node ID: 1001, Rute ID: 10, Distance: 100']
```

Figura 1: Rutas Fijas

```
Order ID: 1, RuteNode: Node ID: 103, Rute ID: 1, Distance: 60, Cantidad: 6000
Order ID: 2, RuteNode: Node ID: 202, Rute ID: 2, Distance: 55, Cantidad: 10000
Order ID: 3, RuteNode: Node ID: 301, Rute ID: 3, Distance: 70, Cantidad: 5000
Order ID: 4, RuteNode: Node ID: 403, Rute ID: 4, Distance: 120, Cantidad: 7000
Order ID: 5, RuteNode: Node ID: 502, Rute ID: 5, Distance: 50, Cantidad: 38000
Order ID: 6, RuteNode: Node ID: 601, Rute ID: 6, Distance: 140, Cantidad: 6000
Order ID: 7, RuteNode: Node ID: 703, Rute ID: 7, Distance: 70, Cantidad: 11000
Order ID: 8, RuteNode: Node ID: 802, Rute ID: 8, Distance: 80, Cantidad: 15000
Order ID: 9, RuteNode: Node ID: 901, Rute ID: 9, Distance: 65, Cantidad: 25000
Order ID: 10, RuteNode: Node ID: 1003, Rute ID: 10, Distance: 40, Cantidad: 9000
Order ID: 11, RuteNode: Node ID: 102, Rute ID: 1, Distance: 45, Cantidad: 10000
Order ID: 12, RuteNode: Node ID: 201, Rute ID: 2, Distance: 60, Cantidad: 9000
Order ID: 13, RuteNode: Node ID: 303, Rute ID: 3, Distance: 50, Cantidad: 21000
Order ID: 14, RuteNode: Node ID: 402, Rute ID: 4, Distance: 90, Cantidad: 6000
Order ID: 15, RuteNode: Node ID: 501, Rute ID: 5, Distance: 75, Cantidad: 24000
Order ID: 16, RuteNode: Node ID: 603, Rute ID: 6, Distance: 120, Cantidad: 5000
Order ID: 17, RuteNode: Node ID: 702, Rute ID: 7, Distance: 60, Cantidad: 20000
Order ID: 18, RuteNode: Node ID: 801, Rute ID: 8, Distance: 50, Cantidad: 11000
Order ID: 19, RuteNode: Node ID: 903, Rute ID: 9, Distance: 75, Cantidad: 20000
Order ID: 20, RuteNode: Node ID: 1002, Rute ID: 10, Distance: 80, Cantidad: 9000
```

Figura 2: Peticiones Fijas

Para extraer el texto de los pdf se hace uso de la biblioteca de Python **fitz**, luego este texto en forma de string se le suman las ordenes específicas para la generación, así como el formato de la clase **Vehicle** para que lo genere en forma de json. Dada la respuesta del modelo, se extrae solo la parte del json haciendo uso de la biblioteca **re**, que es donde están las variables generadas, para luego haciendo uso de la biblioteca **json** de Python acceder a la información generada, para crear las variables a partir de la clase **Vehicle**, esto se hace con el método **obtain vehicles(cant:int=None)** el cual puede recibir como parametro un entero, el cual especificará en el contexto la cantidad de vehículos a generar. Ejemplo, la ejecución del método pasándole como parámetro 10, genera los siguientes 10 vehículos:

```
Vehicle ID: 1, Altura: 3.5, Capacidad: 5000.0, TipoDeCombustible: ['gasolina'], DistMax: 500.0
Vehicle ID: 2, Altura: 3.8, Capacidad: 7500.0, TipoDeCombustible: ['diesel'], DistMax: 475.0
Vehicle ID: 3, Altura: 4.0, Capacidad: 10000.0, TipoDeCombustible: ['gasolina', 'diesel'], DistMax: 450.0
Vehicle ID: 4, Altura: 3.2, Capacidad: 6000.0, TipoDeCombustible: ['gas'], DistMax: 490.0
Vehicle ID: 5, Altura: 4.2, Capacidad: 25000.0, TipoDeCombustible: ['diesel'], DistMax: 300.0
Vehicle ID: 6, Altura: 3.9, Capacidad: 15000.0, TipoDeCombustible: ['gasolina'], DistMax: 400.0
Vehicle ID: 7, Altura: 3.6, Capacidad: 8000.0, TipoDeCombustible: ['hidrogeno'], DistMax: 470.0
Vehicle ID: 8, Altura: 4.1, Capacidad: 30000.0, TipoDeCombustible: ['gasolina', 'diesel'], DistMax: 250.0
Vehicle ID: 9, Altura: 3.7, Capacidad: 9000.0, TipoDeCombustible: ['gas'], DistMax: 460.0
Vehicle ID: 10, Altura: 4.3, Capacidad: 35000.0, TipoDeCombustible: ['diesel'], DistMax: 200.0
```

Figura 3: Vehículos generados con parámetro cantidad

Al poder generar vehículos solo resta verificar que para los vehículos generados se encuentra una solución, o conjunto de asignación que no deje ningún pedido sin asignarle un vehículo. Para ello se implementó la clase **System**, el cual contiene las rutas, los pedidos diarios y los vehículos generados, en su inicialización se determina para cada ruta, el dominio de vehículos que pueden ser asignados a clientes que pertenecen a dichas rutas, así como para cada pedido, se determina el dominio de vehículos que pueden ser asignados, que además de estar restringido a la ruta a la que pertenece el pedido, también está restringido por la cantidad que admite el vehículo, ya que puede darse el caso en que el pedido tenga como cantidad, similar a la que admite la ruta, sin embargo uno de los vehículos del dominio tiene capacidad máxima menor que la del pedido, y menor que la de la ruta, perteneciendo así al dominio de la ruta, pero no al dominio del pedido.

Al tener el dominio para cada pedido se puede probar para cada posible valor haciendo backtrack, sin embargo si se organizan los dominios y variables teniendo en cuenta algunos aspectos, se puede llegar a la solución más rápidamente, o llegar a que no hay solución más rápidamente. Por ejemplo se pueden organizar los pedidos, de manera que se comience a asignar a las variables que tengan menos elementos en el dominio, ya que así se evitaría revisar muchos casos. Otro ejemplo sería organizar el propio dominio, ejemplo, dado el dominio de cierto elemento, elegir el vehículo que pueda asignarse a menos variables o pedidos, ya que si las pocas variables que pueden admitir dichos vehículos admiten a otros, puede ocurrir que estos vehículos se queden sin acceder a ningún pedido, corriendo el riesgo de que se quede algún pedido sin asignarle un vehículo. Otra manera de organizarlo puede ser para explotar al máximo los vehículos, donde se organiza por cantidad de tiempo restante, donde se garantiza que los vehículos den todos los viajes que puedan, o lo que es lo mismo satisfagan la mayor cantidad de pedidos posibles, lo que pudiese minimizar la cantidad de vehículos usados. Y otra forma también sería organizarlo al igual que el anterior pero con tiempo máximo.

Ahora nótese que con generar 10 vehículos no es suficiente para cubrir la necesidad, ya que al ejecutar las asignaciones como problema de satisfacción de

restricciones, devuelve solución vacía, para mejorar esto se creará un método que irá aumentando gradualmente la cantidad de vehículos a presentar, hasta encontrar una solución inicial no vacía. Luego de ejecutar múltiples iteraciones, se encontró una solución inicial para una cantidad de 50 vehículos, que son los siguientes:

```
Vehicle ID: 1, Altura: 3.5, Capacidad: 25000.0, TipoDeCombustible: ['gasolina', 'diesel'], DistMax: 300.0
Vehicle ID: 2, Altura: 3.2, Capacidad: 10000.0, TipoDeCombustible: ['diesel'], DistMax: 450.0
Vehicle ID: 3, Altura: 3.8, Capacidad: 30000.0, TipoDeCombustible: ['gasolina'], DistMax: 250.0
Vehicle ID: 4, Altura: 2.8, Capacidad: 5000.0, TipoDeCombustible: ['gas'], DistMax: 500.0
Vehicle ID: 5, Altura: 4.0, Capacidad: 40000.0, TipoDeCombustible: ['gasolina', 'diesel', 'gas'], DistMax: 150.0
Vehicle ID: 6, Altura: 3.3, Capacidad: 15000.0, TipoDeCombustible: ['diesel'], DistMax: 400.0
Vehicle ID: 7, Altura: 3.6, Capacidad: 22000.0, TipoDeCombustible: ['gasolina'], DistMax: 330.0
Vehicle ID: 8, Altura: 2.9, Capacidad: 7000.0, TipoDeCombustible: ['gas'], DistMax: 480.0
Vehicle ID: 9, Altura: 3.9, Capacidad: 35000.0, TipoDeCombustible: ['gasolina', 'diesel'], DistMax: 200.0
Vehicle ID: 10, Altura: 3.1, Capacidad: 8000.0, TipoDeCombustible: ['diesel'], DistMax: 470.0
Vehicle ID: 11, Altura: 3.7, Capacidad: 28000.0, TipoDeCombustible: ['gasolina'], DistMax: 270.0
Vehicle ID: 12, Altura: 2.7, Capacidad: 4000.0, TipoDeCombustible: ['gas'], DistMax: 510.0
Vehicle ID: 13, Altura: 4.1, Capacidad: 45000.0, TipoDeCombustible: ['gasolina', 'diesel', 'gas'], DistMax: 100.0
Vehicle ID: 14, Altura: 3.4, Capacidad: 12000.0, TipoDeCombustible: ['diesel'], DistMax: 430.0
Vehicle ID: 15, Altura: 3.5, Capacidad: 20000.0, TipoDeCombustible: ['gasolina'], DistMax: 350.0
Vehicle ID: 16, Altura: 3.0, Capacidad: 6000.0, TipoDeCombustible: ['gas'], DistMax: 490.0
Vehicle ID: 17, Altura: 3.8, Capacidad: 32000.0, TipoDeCombustible: ['gasolina', 'diesel'], DistMax: 230.0
Vehicle ID: 18, Altura: 3.2, Capacidad: 9000.0, TipoDeCombustible: ['diesel'], DistMax: 460.0
Vehicle ID: 19, Altura: 3.6, Capacidad: 24000.0, TipoDeCombustible: ['gasolina'], DistMax: 310.0
Vehicle ID: 20, Altura: 2.8, Capacidad: 5500.0, TipoDeCombustible: ['gas'], DistMax: 495.0
Vehicle ID: 21, Altura: 3.9, Capacidad: 38000.0, TipoDeCombustible: ['gasolina', 'diesel', 'gas'], DistMax: 170.0
Vehicle ID: 22, Altura: 3.3, Capacidad: 13000.0, TipoDeCombustible: ['diesel'], DistMax: 420.0
Vehicle ID: 23, Altura: 3.7, Capacidad: 26000.0, TipoDeCombustible: ['gasolina'], DistMax: 290.0
Vehicle ID: 24, Altura: 2.9, Capacidad: 6500.0, TipoDeCombustible: ['gas'], DistMax: 485.0
Vehicle ID: 25, Altura: 4.0, Capacidad: 42000.0, TipoDeCombustible: ['gasolina', 'diesel'], DistMax: 130.0
Vehicle ID: 26, Altura: 3.1, Capacidad: 7500.0, TipoDeCombustible: ['diesel'], DistMax: 475.0
Vehicle ID: 27, Altura: 3.8, Capacidad: 29000.0, TipoDeCombustible: ['gasolina'], DistMax: 260.0
Vehicle ID: 28, Altura: 2.7, Capacidad: 4500.0, TipoDeCombustible: ['gas'], DistMax: 505.0
Vehicle ID: 29, Altura: 4.1, Capacidad: 48000.0, TipoDeCombustible: ['gasolina', 'diesel', 'gas'], DistMax: 70.0
Vehicle ID: 30, Altura: 3.4, Capacidad: 11000.0, TipoDeCombustible: ['diesel'], DistMax: 440.0
Vehicle ID: 31, Altura: 3.5, Capacidad: 19000.0, TipoDeCombustible: ['gasolina'], DistMax: 360.0
Vehicle ID: 32, Altura: 3.0, Capacidad: 6800.0, TipoDeCombustible: ['gas'], DistMax: 482.0
Vehicle ID: 33, Altura: 3.8, Capacidad: 33000.0, TipoDeCombustible: ['diesel', 'gasolina'], DistMax: 220.0
Vehicle ID: 34, Altura: 3.2, Capacidad: 8500.0, TipoDeCombustible: ['diesel'], DistMax: 465.0
Vehicle ID: 35, Altura: 3.6, Capacidad: 23000.0, TipoDeCombustible: ['gasolina'], DistMax: 320.0
Vehicle ID: 36, Altura: 2.8, Capacidad: 5200.0, TipoDeCombustible: ['gas'], DistMax: 498.0
Vehicle ID: 37, Altura: 3.9, Capacidad: 37000.0, TipoDeCombustible: ['gasolina', 'diesel', 'gas'], DistMax: 180.0
Vehicle ID: 38, Altura: 3.3, Capacidad: 14000.0, TipoDeCombustible: ['diesel'], DistMax: 410.0
Vehicle ID: 39, Altura: 3.7, Capacidad: 27000.0, TipoDeCombustible: ['gasolina'], DistMax: 280.0
Vehicle ID: 40, Altura: 2.9, Capacidad: 7200.0, TipoDeCombustible: ['gas'], DistMax: 478.0
Vehicle ID: 41, Altura: 4.0, Capacidad: 41000.0, TipoDeCombustible: ['gasolina', 'diesel'], DistMax: 140.0
Vehicle ID: 42, Altura: 3.1, Capacidad: 7800.0, TipoDeCombustible: ['diesel'], DistMax: 472.0
Vehicle ID: 43, Altura: 3.8, Capacidad: 28500.0, TipoDeCombustible: ['gasolina'], DistMax: 265.0
Vehicle ID: 44, Altura: 2.7, Capacidad: 4200.0, TipoDeCombustible: ['gas'], DistMax: 508.0
Vehicle ID: 45, Altura: 4.1, Capacidad: 47000.0, TipoDeCombustible: ['gasolina', 'diesel', 'gas'], DistMax: 80.0
Vehicle ID: 46, Altura: 3.4, Capacidad: 11500.0, TipoDeCombustible: ['diesel'], DistMax: 435.0
Vehicle ID: 47, Altura: 3.5, Capacidad: 19500.0, TipoDeCombustible: ['gasolina'], DistMax: 355.0
Vehicle ID: 48, Altura: 3.0, Capacidad: 6300.0, TipoDeCombustible: ['gas'], DistMax: 487.0
Vehicle ID: 49, Altura: 3.8, Capacidad: 32500.0, TipoDeCombustible: ['diesel', 'gasolina'], DistMax: 225.0
Vehicle ID: 50, Altura: 3.2, Capacidad: 8200.0, TipoDeCombustible: ['diesel'], DistMax: 468.0
```

Figura 4: Vehículos generados con parámetro cantidad=50

Organizando el dominio de manera que selecciono primero los vehículos que pueden satisfacer menos pedidos se obtuvo el siguiente resultado:

```

Order 5 assigned to Vehicle 25
Order 15 assigned to Vehicle 5
Order 9 assigned to Vehicle 9
Order 19 assigned to Vehicle 15
Order 8 assigned to Vehicle 6
Order 13 assigned to Vehicle 7
Order 17 assigned to Vehicle 15
Order 7 assigned to Vehicle 6
Order 20 assigned to Vehicle 18
Order 4 assigned to Vehicle 8
Order 6 assigned to Vehicle 16
Order 2 assigned to Vehicle 2
Order 18 assigned to Vehicle 6
Order 11 assigned to Vehicle 2
Order 12 assigned to Vehicle 18
Order 10 assigned to Vehicle 18
Order 16 assigned to Vehicle 4
Order 14 assigned to Vehicle 16
Order 1 assigned to Vehicle 8
Order 3 assigned to Vehicle 4
Cantidad de vehiculos usados: 11
cantidad de back dados: 13

```

Figura 5: Resultado 1

Organizando el dominio priorizando los que tengan menor distancia restante, o menor cantidad de viajes por dar se obtuvo el resultado siguiente:

```

Order 5 assigned to Vehicle 25
Order 15 assigned to Vehicle 5
Order 9 assigned to Vehicle 9
Order 19 assigned to Vehicle 33
Order 8 assigned to Vehicle 6
Order 13 assigned to Vehicle 3
Order 17 assigned to Vehicle 1
Order 7 assigned to Vehicle 1
Order 20 assigned to Vehicle 30
Order 4 assigned to Vehicle 10
Order 6 assigned to Vehicle 16
Order 2 assigned to Vehicle 2
Order 18 assigned to Vehicle 6
Order 11 assigned to Vehicle 6
Order 12 assigned to Vehicle 2
Order 10 assigned to Vehicle 2
Order 16 assigned to Vehicle 20
Order 14 assigned to Vehicle 16
Order 1 assigned to Vehicle 2
Order 3 assigned to Vehicle 3
Cantidad de vehiculos usados: 12
cantidad de back dados: 10

```

Figura 6: Resultado 2

Organizando el dominio priorizando los que tengan menor distancia máxima, o los que tengan menor cantidad de viajes, que primero selecciona los vehículos que tienen mayor capacidad, se obtiene el siguiente resultado

```
Order 5 assigned to Vehicle 25
Order 15 assigned to Vehicle 5
Order 9 assigned to Vehicle 9
Order 19 assigned to Vehicle 33
Order 8 assigned to Vehicle 6
Order 13 assigned to Vehicle 3
Order 17 assigned to Vehicle 1
Order 7 assigned to Vehicle 1
Order 20 assigned to Vehicle 30
Order 4 assigned to Vehicle 10
Order 6 assigned to Vehicle 16
Order 2 assigned to Vehicle 2
Order 18 assigned to Vehicle 6
Order 11 assigned to Vehicle 6
Order 12 assigned to Vehicle 2
Order 10 assigned to Vehicle 30
Order 16 assigned to Vehicle 20
Order 14 assigned to Vehicle 10
Order 1 assigned to Vehicle 38
Order 3 assigned to Vehicle 3
Cantidad de vehiculos usados: 13
cantidad de back dados: 5
```

Figura 7: Resultado 3

Puede apreciarse que con la primera forma es como se logra utilizar menor cantidad de vehículos, sin embargo es el que más explora las soluciones, ya que tiene que probar más casos, por otro lado está el que explota los vehículos que ya están siendo utilizado, el cual usa más vehículos que el anterior orden, sin embargo explora menor cantidad de soluciones, y por último,