

# Proyecto Seneca-Libre: Etapa 2 - Implementación y Resolución del Modelo de Ruteo Vehicular

## **Instructores:**

Carlos Andrés Lozano,  
Germán Adolfo Montoya,  
Juan Andrés Mendez

## **Abstract**

Este documento describe la segunda etapa del proyecto de optimización logística de Seneca Libre, enfocada en la implementación y validación del modelo matemático desarrollado previamente para mejorar la eficiencia en las rutas de suministro. El objetivo de esta fase es demostrar que el modelo propuesto refleja con precisión la operación logística real, permitiendo la predicción de rutas y costos de forma confiable. A través de la implementación en entornos de programación y la resolución de diversos escenarios de prueba, se espera obtener datos que permitan evaluar la viabilidad y precisión del modelo. Esta fase incluye un análisis detallado de los resultados, destacando las oportunidades de mejora en la eficiencia operativa y la reducción de costos. Con esta validación, Seneca Libre busca no solo consolidarse como líder en el mercado de comercio electrónico, sino también establecer una base para futuras simulaciones y ajustes que impulsen la sostenibilidad y competitividad de sus operaciones.

## **1 Introducción**

En Seneca Libre, nuestra misión es mantenernos como líderes en el comercio electrónico en Bogotá mediante una operación logística eficiente y altamente competitiva. En la primera etapa de este proyecto, desarrollamos un modelo matemático integral para optimizar nuestras rutas de suministro, abordando los desafíos operacionales que enfrentamos debido a la alta demanda, los costos elevados, y las limitaciones en nuestra flota de más de 700 vehículos. Este modelo fue diseñado para reducir costos operativos, mejorar la puntualidad de las entregas y optimizar la carga de trabajo de nuestros transportistas.

Con el modelado inicial completado, avanzamos ahora a la etapa de implementación. En esta fase, el objetivo es demostrar que el modelo diseñado representa de forma precisa nuestra operación logística y permite predecir los costos y rutas de manera realista. Cada equipo deberá implementar el modelo en un entorno de programación (Python con Pyomo o PuLP, o GAMS), y resolver una serie de casos de prueba que reflejan diferentes escenarios operacionales. A través de esta implementación, esperamos obtener datos valiosos sobre las rutas y los costos óptimos que el modelo es capaz de proponer, lo que servirá para evaluar su viabilidad y precisión.

La importancia de esta etapa radica en la validación y análisis de los resultados obtenidos, los cuales deben ofrecer una visión clara de las mejoras potenciales en la

eficiencia de nuestras operaciones. Los equipos deberán presentar no solo los resultados de las rutas optimizadas, sino también un análisis detallado de los costos involucrados, identificando cualquier posible discrepancia entre el modelo y las operaciones reales. La información recopilada permitirá a Seneca Libre tomar decisiones informadas sobre futuras implementaciones y ajustes en la estrategia logística.

Esta segunda entrega no solo busca validar el modelo propuesto, sino también establecer una base sólida para futuras simulaciones y adaptaciones. Con ello, Seneca Libre pretende no solo mantenerse al frente del mercado, sino también reducir significativamente los costos y tiempos de entrega, elevando los estándares de servicio para nuestros clientes y mejorando nuestra sostenibilidad en el competitivo mundo del comercio electrónico.

## 2 Objetivos Específicos de la Etapa 2

- Implementar el modelo matemático en un lenguaje de programación (*Python* con *Pyomo* o *PuLP*, o *GAMS*).
- Resolver varios casos de prueba para analizar el comportamiento del modelo y validar su efectividad en la optimización de costos.
- Explorar escenarios operacionales avanzados, incluyendo:
  - La incorporación de **nodos de recarga** en la optimización con *Pyomo*.
  - La resolución de una versión simplificada del problema utilizando **algoritmos genéticos** como estrategia heurística.
  - La **simulación de procesos de ruteo**, evaluando el impacto de distintas configuraciones y decisiones en escenarios dinámicos.
- Realizar análisis de resultados para optimizar la flota y reducir el tiempo de entrega.

## 3 Datos y Recursos Disponibles

Para la implementación y resolución del modelo de ruteo vehicular en esta etapa, se cuenta con diversos datos de entrada que describen los puntos de entrega, centros de distribución, vehículos y nodos de recarga. Aunque algunos datos adicionales, como los tipos de productos y las capacidades específicas de los centros de distribución, no son críticos para los primeros casos de prueba, se incluyen a continuación para una referencia completa.

### 3.1 Datos de Puntos de Entrega

Cada cliente se representa mediante coordenadas geográficas, utilizando la latitud y longitud para determinar su ubicación en el mapa. El conjunto de datos incluye:

- **NodeID:** Un identificador único para cada nodo. (Puede ser un cliente o un centro de distribución o un nodo de recarga).
- **ClientID:** Un identificador único para cada cliente.

- **Demanda**[ $kg$ ]: Cantidad de producto solicitada por el cliente, expresada en kilogramos (kg). (\* **Nota:** Para uno de los casos donde hay multiples productos, se debe considerar la demanda de cada producto por separado i.e ‘Producto-Type-A’, ‘Producto-Type-B’, ‘Producto-Type-C’)
- **Longitud y Latitud:** Coordenadas geográficas en grados decimales, expresadas como  $Longitud_{cliente}$  ( $^{\circ}$ ) y  $Latitud_{cliente}$  ( $^{\circ}$ ).

### 3.2 Datos de Centros de Distribución

Los centros de distribución (o "depots") se representan con su ubicación geográfica y la capacidad de almacenamiento por tipo de producto:

- **NodeID:** Un identificador único para cada nodo. (Puede ser un cliente o un centro de distribución o un nodo de recarga).
- **DepotID:** Un identificador único para cada centro de distribución.
- **Longitud y Latitud:** Coordenadas geográficas en grados decimales, expresadas como  $Longitud_{depot}$  ( $^{\circ}$ ) y  $Latitud_{depot}$  ( $^{\circ}$ ).
- **Capacidad de Producto**[ $kg$ ]: Capacidad de almacenamiento por cada tipo de producto (\* **Nota** A, B y C Dependiendo si tenemos uno mas productos.) en cada centro de distribución, expresada en unidades de producto. Aunque se incluyen en el conjunto de datos, estas capacidades no son relevantes para los primeros casos de prueba.

### 3.3 Datos de Vehículos

Los datos de vehículos disponibles incluyen información detallada sobre el tipo de vehículo, capacidad de carga, rango operativo y costos asociados. A continuación, se describen los parámetros específicos de cada tipo de vehículo:

- **VehicleType:** Tipo de vehículo. Se incluyen tres tipos:
  - *Gas Car:* Vehículo de gasolina con parámetros específicos de eficiencia y costos.
  - *Drone:* Vehículo aéreo no tripulado que utiliza electricidad como fuente de energía.
  - *Solar EV:* Vehículo eléctrico con capacidad de recarga solar.
- **Capacidad** ( $C_{vehículo}$ ): Capacidad máxima de carga de cada vehículo, expresada en kilogramos (kg). Cada vehículo tiene una capacidad específica para transportar carga.
- **Rango** ( $R_{vehículo}$ ): Distancia máxima que puede recorrer el vehículo con una carga completa de combustible o energía, expresada en kilómetros (km).
- **Tarifa de Flete** (Freight Rate): Costo por kilómetro recorrido por cada tipo de vehículo, expresado en pesos colombianos por kilómetro (COP/km).
- **Tarifa de Tiempo** (Time Rate): Costo por minuto de operación del vehículo, expresado en pesos colombianos por minuto (COP/min).

- **Costo de Mantenimiento Diario** (Daily Maintenance): Costo diario de mantenimiento del vehículo, expresado en pesos colombianos por día (COP/día).
- **Costo de Recarga/Combustible** (Recharge/Fuel Cost): Costo de recarga o combustible por unidad (galón o kWh), expresado en pesos colombianos (COP/gal o COP/kWh).
- **Tiempo de Recarga/Combustible** (Recharge/Fuel Time): Tiempo necesario para recargar un 10% de la capacidad total de energía o combustible del vehículo, expresado en minutos (min/10% de carga).
- **Velocidad Promedio** (Avg. Speed): Velocidad promedio del vehículo, expresada en kilómetros por hora (km/h). (\* **Nota:** Este solo esta incluido para los drones ya que para los vehiculos terrestres la velocidad promedio no es la más importante.)
- **Eficiencia de Combustible** (Gas Efficiency): Eficiencia del vehículo en cuanto a consumo de gasolina, expresada en kilómetros por galón (km/gal) para vehículos de gasolina.
- **Eficiencia Energética** (Electricity Efficiency): Eficiencia en consumo de electricidad, expresada en kilovatios hora por kilómetro (kWh/km) para vehículos eléctricos y drones.

Formula para hallar el combustible o energia que se necesita para recorrer una distancia  $d$  en un vehiculo:

$$\text{Combustible/Energía} = \frac{d}{\text{Eficiencia de Combustible/Energía}} \quad (1)$$

Formula para hallar el tiempo que se necesita para recargar el combustible o energia que se necesita para recorrer una distancia  $d$  en un vehiculo:

$$\text{Capacidad Total de Energía o Combustible} = \frac{\text{Rango del Vehículo}}{\text{Eficiencia del Vehículo}} \quad (2)$$

$$\text{Energía o Combustible al 10\%} = 0.1 \times \text{Capacidad Total de Energía o Combustible} \quad (3)$$

$$\text{Tiempo de Recarga o Reabastecimiento} = \frac{\text{Energía o Combustible al 10\%}}{\text{Tasa de Recarga o Reabastecimiento}} \quad (4)$$

Los datos específicos para cada tipo de vehículo se detallan en la siguiente tabla:

### 3.3.1 Velocidad de Carga

Además de los costos de operación y recarga, el proceso de carga de cada vehículo tiene un costo asociado. Cada minuto de carga tiene un costo de 500 COP, y se puede cargar un promedio de 5 kg de carga por minuto, en función del trabajo de un empleado encargado de la carga de los vehículos.

Definimos el costo total de carga ( $C_{\text{carga}}$ ) para un vehículo de la siguiente forma:

$$C_{\text{carga}} = \text{Carga} \times \frac{500 \text{ COP}}{\text{kg/min}} \quad (5)$$

Parámetro	Gas Car	Drone	Solar EV
Freight Rate [COP/km]	5000	500	4000
Time Rate [COP/min]	500	500	500
Daily Maintenance [COP/day]	30000	3000	21000
Recharge/Fuel Cost [COP/(gal or kWh)]	16000	220.73	-
Recharge/Fuel Time [min/10% charge]	0.1	2	-
Avg. Speed [km/h]	-	40	-
Gas Efficiency [km/gal]	10	-	-
Electricity Efficiency [kWh/km]	-	0.15	0.15

Table 1: Parámetros de Costos y Eficiencia para Vehículos

donde *Carga* es la cantidad de peso en kilogramos (kg) que se carga en el vehículo en un tiempo determinado.

### 3.4 Datos de Nodos de Recarga

Los nodos de recarga representan estaciones donde los vehículos pueden recargar combustible o energía. Estos nodos están definidos por:

- **NodeID:** Un identificador único para cada nodo. (Puede ser un cliente o un centro de distribución).
- **NodeID:** Un identificador único para cada nodo de recarga.
- **Longitud y Latitud:** Coordenadas geográficas en grados decimales, expresadas como  $\text{Longitud}_{\text{nodo}}$  (°) y  $\text{Latitud}_{\text{nodo}}$  (°).

### 3.5 Datos de Costos

Para el modelo de ruteo vehicular, los costos se componen principalmente de los gastos asociados a las distancias y tiempos operativos de cada vehículo. Aunque algunos datos de costos específicos están directamente relacionados con el tipo de vehículo, como las tarifas de flete y tiempo, otros deben calcularse en función de la distancia recorrida y el tiempo empleado en cada ruta.

El costo operativo de cada vehículo en una ruta particular se puede expresar en función de la distancia ( $d$ , en km) y el costo por kilómetro ( $c_{\text{km}}$ ), obteniendo el costo total ( $C_{\text{total}}$ ) mediante la siguiente ecuación:

$$C_{\text{total}} = c_{\text{km}} \times d \quad (6)$$

donde  $c_{\text{km}}$  es el costo por kilómetro de operación para el tipo específico de vehículo utilizado, y  $d$  representa la distancia total recorrida en kilómetros.

De igual manera, se debe considerar el tiempo de operación ( $t$ , en minutos) y el costo por minuto ( $c_{\text{min}}$ ) asociado a cada vehículo para obtener el costo total en función del tiempo:

$$C_{\text{tiempo}} = c_{\text{min}} \times t \quad (7)$$

donde  $c_{\text{min}}$  es el costo por minuto para el vehículo y  $t$  es el tiempo total de la ruta en minutos.

Por lo tanto, el costo total de operación ( $C_{\text{operativo}}$ ) para una ruta puede calcularse como la suma de los costos de distancia y tiempo:

$$C_{\text{operativo}} = C_{\text{total}} + C_{\text{tiempo}} \quad (8)$$

Estos valores deberán ser calculados según las rutas específicas de cada caso de prueba, ya que las distancias y tiempos varían en cada escenario. Los demás costos, tales como mantenimiento diario, tarifas de recarga y costos asociados al proceso de carga, ya están cubiertos en los datos específicos del vehículo y no requieren cálculos adicionales en esta fase del análisis.

## 3.6 Herramientas de Ayuda, Recursos Adicionales y Consideraciones

Para el desarrollo del proyecto, se recomienda utilizar herramientas avanzadas de optimización que permitan resolver el modelo de ruteo vehicular de manera eficiente, especialmente en los casos de mayor complejidad y escalabilidad. A continuación, se presentan algunos de los solvers más utilizados y recursos adicionales para su instalación y configuración con Python y Pyomo.

### 3.6.1 Herramientas de Ayuda

A continuación se presentan algunas herramientas clave para resolver el modelo de ruteo vehicular de manera eficiente, especialmente en los casos de mayor complejidad y escalabilidad. Estas incluyen solvers de optimización de gran escala y recursos para la generación de datos sintéticos que ayudarán en el desarrollo y análisis del proyecto.

#### Solvers de Optimización a Gran Escala:

- **Gurobi:** Gurobi es uno de los solvers comerciales más potentes y ampliamente utilizados en optimización a gran escala. Soporta modelos lineales, no lineales y enteros mixtos. Es reconocido por su alta eficiencia y velocidad en problemas complejos. Para aprender a integrarlo con marcos de modelado de código abierto como Pyomo, consulte el siguiente tutorial: <https://support.gurobi.com/hc/en-us/articles/16316467014801-How-do-I-use-Gurobi-with-open-source-modeling-frameworks>. Puede encontrar más información y recursos adicionales en su sitio web oficial: <https://www.gurobi.com/>.
- **CPLEX:** CPLEX, desarrollado por IBM, es otro solver de alto rendimiento ampliamente utilizado en optimización a gran escala. Ofrece soporte para una variedad de problemas complejos, incluyendo modelos lineales, no lineales y enteros mixtos. Para instalar y utilizar CPLEX con Pyomo, consulte el siguiente recurso: <https://eseslab.com/Downloads/Installing-PYOMO-and-CPLEX.pdf>. Puede acceder a la página oficial de CPLEX para información adicional: <https://www.ibm.com/analytics/cplex-optimizer>.
- **HiGHS:** HiGHS es un solver de código abierto y eficiente para resolver problemas de optimización lineal y entera mixta. En este caso, es el solver preferido debido a su ligereza y accesibilidad. Sin embargo, presenta algunas limitaciones, como la cantidad de nodos que puede evaluar simultáneamente. Existen tres métodos principales para instalarlo e integrarlo:

1. **\*\*Mediante el ejecutable:\*\*** Aunque esta es una opción viable, actualmente el ejecutable de HiGHS no es reconocido directamente por **Pyomo**, lo que requiere realizar configuraciones adicionales para que funcione correctamente.
2. **\*\*Con `appsi`:** Esta integración directa desde Pyomo es la más recomendada para usuarios de **Mac** y **Linux**, ya que simplifica considerablemente el proceso de configuración.
3. **\*\*Con `amplpy`:** Para usuarios de **Windows**, esta opción resulta la más sencilla, ya que **amplpy** permite una integración rápida y eficiente con HiGHS.

En el repositorio que se les proporcionará, encontrarán un tutorial detallado que explica cómo probar cada una de estas rutas para integrar HiGHS en sus proyectos. Es importante tener instalado un solver más avanzado, como Gurobi o CPLEX, para manejar problemas de mayor complejidad o con un número significativo de nodos, ya que HiGHS puede ser limitado en estas circunstancias. Recursos adicionales para integrar HiGHS con Python y Pyomo están disponibles en: <https://ergo-code.github.io/HiGHS/dev/interfaces/python/>, y más detalles técnicos pueden encontrarse en su página oficial: <https://highs.dev/>.

**Generador de Datos Sintéticos:** Para ayudar en la creación de datos representativos y personalizados según los requisitos del problema, se ha proporcionado un generador de datos sintéticos. Este recurso permite generar datos similares a los de los casos de prueba, facilitando la experimentación con diferentes configuraciones sin necesidad de recolectar datos reales.

El generador de datos y los casos de prueba, tanto **estándares** como **especiales**, estarán ubicados en un repositorio de **GitHub**, el cual se compartirá con todos los estudiantes. Este repositorio centralizará las herramientas y recursos necesarios para el desarrollo del proyecto.

Además, en este repositorio se incluirá un *notebook* con todos los costos y parámetros relevantes de los vehículos, centros de distribución y clientes. Este archivo servirá como referencia clave para integrar el generador de datos con los modelos de optimización.

Estas herramientas están diseñadas para optimizar el proceso de modelado y análisis, permitiendo a los estudiantes enfocarse en el desarrollo y prueba de su modelo de ruteo vehicular de manera eficiente y efectiva.

### 3.6.2 Recursos Adicionales

**Open Source Routing Machine (OSRM):** OSRM es un motor de enrutamiento de código abierto que ofrece una API potente para calcular rutas óptimas, tiempos de viaje y distancias entre puntos. Este recurso es fundamental para generar las matrices de costos para vehículos terrestres, ya que permite incorporar de manera realista las restricciones de infraestructura vial y las características de las rutas, como distancias reales y tiempos de viaje, que no pueden ser modelados con precisión usando simples cálculos de distancia euclidiana.

Para más información y acceso a la documentación oficial de OSRM, visite su sitio web: <http://project-osrm.org/>. Adicionalmente, puede aprender a utilizar OSRM y generar visualizaciones de rutas mediante el siguiente tutorial: <https://medium.com/walmartglobaltech/finding-and-plotting-optimal-route-using-open-source-api-in-python>.

Este tutorial explica cómo obtener la distancia, la duración del viaje y la nube de puntos para representar gráficamente la ruta.

**Por qué utilizar OSRM para Vehículos Terrestres** A diferencia de los drones, los vehículos terrestres están sujetos a las restricciones de la infraestructura vial, como calles, intersecciones y límites de velocidad. Usar una API como OSRM permite calcular distancias y tiempos de viaje basados en la red vial real, en lugar de aproximaciones inexactas como la distancia euclidiana o haversiana. Esto es crucial para obtener una matriz de costos representativa que permita optimizar rutas de manera más efectiva y realista.

**Uso de OSRM para la Generación de Matrices de Costos** La API de OSRM debe ser utilizada para calcular las distancias y tiempos de viaje entre los nodos de clientes y los centros de distribución. Con esta información se generará la matriz de costos, tanto para vehículos de gasolina como eléctricos. Esta matriz será clave para la optimización de rutas basada en los costos operativos específicos de cada tipo de vehículo.

El tutorial compartido muestra cómo obtener, además de la distancia y la duración del viaje, la nube de puntos (*polyline*) que permite graficar las rutas en un mapa. Sin embargo, para propósitos de optimización y cálculo de la matriz de costos, no es necesario generar la nube de puntos en esta etapa. La idea es primero resolver el problema de optimización y, posteriormente, utilizar OSRM nuevamente para calcular y almacenar únicamente la nube de puntos para las rutas resultantes del problema resuelto. Este enfoque mejora la eficiencia al evitar el almacenamiento y procesamiento innecesario de datos intermedios.

**Cálculo de la Matriz de Costos para Drones** En el caso de los drones, no se ven afectados por las restricciones de infraestructura vial, por lo que se recomienda utilizar la distancia haversiana, una medida precisa de distancia entre dos puntos en la superficie terrestre, junto con la velocidad promedio del dron. Esto permite estimar de manera eficiente los costos de viaje sin depender de APIs externas, lo que simplifica la generación de la matriz de costos para estos vehículos aéreos.

### 3.7 Consideraciones Adicionales

Para garantizar que todos los estudiantes tengan la oportunidad de desarrollar e implementar su modelo de ruteo vehicular de manera efectiva, se brinda flexibilidad para simplificar o modificar ciertos aspectos del problema. Estas modificaciones deben estar justificadas y mantener una relación realista entre los datos y porcentajes relevantes, asegurando la coherencia y la posibilidad de comparar las implementaciones con las de sus compañeros.

A continuación, se presentan las consideraciones clave que deben tener en cuenta:

- **Simplificación de Múltiples Vehículos:** Los estudiantes pueden optar por simplificar el problema de múltiples vehículos a uno con un único tipo de vehículo, siempre y cuando la relación entre capacidades, costos y restricciones operativas se mantenga realista. La coherencia en los datos proporcionados debe justificarse y documentarse en el análisis del modelo.
- **Adaptación de Múltiples Centros de Distribución (Depots):** Es posible modelar el problema desde una perspectiva de un solo centro de distribución (*single*



*depot*) o múltiples centros de distribución (*multi-depot*). Los datos proporcionados incluirán las ubicaciones sugeridas para los centros en ambos escenarios, permitiendo que cada grupo elija y justifique su enfoque.

- **Ajustes en los Costos y Parámetros Operativos:** Los costos y otros parámetros, como tiempos y capacidades, pueden ajustarse según las necesidades del modelo, siempre que las modificaciones se expliquen adecuadamente. Esto incluye mantener proporciones razonables entre los diferentes costos operativos, asegurando que el modelo refleje decisiones viables y comparables.
- **Demanda y Ubicación de los Clientes (Inamovible):** Lo único que debe mantenerse fijo en todos los modelos es la **demanda** de los clientes y la **ubicación geográfica** de cada uno. Estos elementos garantizan que las soluciones sean comparables y que los resultados reflejen el mismo problema base.
- **Experimentación y Justificación:** Los estudiantes son libres de experimentar con simplificaciones o modificaciones en su modelado, siempre que estas sean justificadas y documentadas. Las decisiones tomadas deben ser consistentes con los objetivos del proyecto y permitir un análisis riguroso de las soluciones propuestas.

Además, los datos proporcionados incluirán opciones para determinar las ubicaciones en un escenario *multi-depot* versus un escenario *single depot*, lo que permitirá a los estudiantes decidir el enfoque más adecuado para su modelo. Estas recomendaciones fomentan la creatividad y flexibilidad en el diseño del modelo, garantizando que las soluciones puedan evaluarse de manera equitativa y que reflejen un balance entre realismo y viabilidad técnica.

### 3.7.1 Uso de Datos de Nodos de Recarga

En los casos donde los vehículos necesiten recargar combustible o energía durante sus rutas, los estudiantes deben incluir los nodos de recarga en el modelo. Los nodos de recarga están identificados en los datos proporcionados y cuentan con coordenadas geográficas. Estos nodos deben ser integrados como puntos intermedios en las rutas, asegurando que los vehículos puedan recargar según sus necesidades de combustible o electricidad, especialmente en escenarios de larga distancia o para vehículos con un rango limitado, como los drones y los vehículos eléctricos.

### 3.7.2 Ajustes en la Matriz de Costos y Restricciones de Distancia

Para el cálculo de la matriz de costos, es importante diferenciar entre los vehículos terrestres y los drones:

- **Vehículos Terrestres (Gas Car, Solar EV):** Se recomienda utilizar la API de OSRM para calcular las distancias y tiempos de viaje entre los nodos de clientes y los centros de distribución. Esto generará una matriz de costos precisa basada en las distancias reales de las rutas vehiculares.
- **Drones:** Para los drones, que no siguen rutas de carretera, se sugiere calcular la matriz de costos utilizando la distancia haversiana, basada en las coordenadas geográficas de los nodos y la velocidad promedio del dron. Esto asegura que los costos sean representativos de la realidad operativa de los drones.

### 3.7.3 Coherencia y Representatividad de los Datos

Es crucial que cualquier ajuste realizado en el modelo respete la coherencia y la representatividad de los datos. Los cambios deben alinearse con las restricciones físicas y ser coherentes con los parámetros definidos para cada tipo de vehículo y depot. Esto incluye:

- La capacidad de carga y el rango operativo de los vehículos deben estar bien definidos y aplicarse adecuadamente en cada caso de prueba.
- Las limitaciones de suministro en cada depot deben respetarse, asegurando que cada uno no exceda su capacidad máxima para cada tipo de producto.

Estas consideraciones adicionales ayudarán a asegurar que el modelo se adapte de manera realista y efectiva a las condiciones operativas de Seneca Libre, reflejando las restricciones logísticas y de recursos en cada escenario del proyecto. El único valor que no es modificable son los clientes, y en dado caso que su modelado actual no sea suficiente para llegar a una solución de los casos estandares o especiales tendran que modificar el modelado hasta que lleguen a una solución.

## 4 Descripción de los Casos de Prueba

Para evaluar el modelo de ruteo vehicular, se han definido escenarios estándar y especiales que reflejan diversas condiciones operativas de *Seneca Libre*. Estos casos permiten comprobar la capacidad del modelo para adaptarse a restricciones y desafíos específicos, al tiempo que fomentan la creatividad y experimentación en la implementación y análisis de resultados.

### 4.1 Escenario Base: Validación de Solución Factible

El escenario base está diseñado para garantizar que el modelo genere una solución válida y coherente.

- Se evaluará que las soluciones no presenten **subrutas** y cumplan con las restricciones de capacidad de los vehículos.
- Se utilizarán datos de demanda fija para los clientes y ubicaciones predefinidas.
- Los estudiantes deben garantizar que el modelo respete todas las restricciones establecidas.

**Objetivo:** Validar que el modelo funcione correctamente y produzca soluciones factibles con restricciones básicas. Este escenario asegura una base sólida para los casos más avanzados.

### 4.2 Escenario 2: Evaluación por Costos

En este caso, se analiza la capacidad del modelo para optimizar el costo de operación.

- Se evaluará el costo total de la función objetivo.
- Los grupos con costos dentro del **percentil 80% más bajo** recibirán la calificación máxima (5.0), mientras que por cada **10 puntos adicionales en el costo**, se reducirá una décima de la nota.
- Se utilizan las mismas ubicaciones y demandas que en el escenario base.

**Objetivo:** Determinar la capacidad del modelo para minimizar costos y premiar las soluciones más eficientes.

### 4.3 Escenario 3: Gestión de Oferta en Centros de Distribución

Este escenario introduce restricciones adicionales relacionadas con la oferta en los centros de distribución.

- Cada centro de distribución tiene una capacidad máxima de producto que puede despachar.
- El modelo debe ajustarse para garantizar que los centros no excedan su capacidad.
- Se evaluará la optimización bajo esta nueva restricción.

**Objetivo:** Adaptar el modelo para manejar restricciones operativas en los centros de distribución y analizar su impacto en las rutas.

## 4.4 Escenario 4: Manejo de Múltiples Productos

En este caso, se introduce la complejidad de manejar múltiples tipos de productos en una sola flota de vehículos.

- Cada cliente tiene demandas específicas para varios tipos de productos.
- Los vehículos deben respetar su capacidad total, pero ahora transportarán productos variados.

**Objetivo:** Evaluar la capacidad del modelo para manejar cargas mixtas y optimizar las rutas considerando las restricciones asociadas.

## 4.5 Casos Especiales

Los casos especiales están diseñados para estudiantes que deseen explorar técnicas avanzadas de optimización, heurísticas y simulación. Estos casos permiten bonificaciones acumulativas que incrementan la nota por encima de 5.0.

### 4.5.1 Caso Especial 1: Nodos de Recarga

Este caso introduce nodos de recarga en las rutas para vehículos eléctricos.

- Se incluyen nodos de recarga estratégicos a lo largo de las rutas.
- Los estudiantes deben optimizar las rutas considerando el tiempo y costo asociados con el uso de los nodos.

**Objetivo:** Incorporar nodos de recarga en el modelo para extender la viabilidad de las rutas de largo alcance.

### 4.5.2 Caso Especial 2: Algoritmos Genéticos

Este caso permite resolver una versión simplificada del problema utilizando algoritmos genéticos.

- Se debe implementar un enfoque heurístico basado en algoritmos genéticos.
- Se evalúa la calidad de la solución y su tiempo de ejecución en comparación con métodos exactos.

**Objetivo:** Aplicar técnicas heurísticas para abordar el problema de ruteo con un enfoque alternativo.

### 4.5.3 Caso Especial 3: Simulación de Procesos de Ruteo

Los estudiantes simularán escenarios dinámicos basados en las soluciones obtenidas.

- Se simula el impacto de distintas configuraciones y decisiones operativas en el tiempo de entrega y la satisfacción del cliente.
- Se utilizarán herramientas de simulación para modelar cambios en la demanda o condiciones externas.

**Objetivo:** Evaluar la robustez y adaptabilidad del modelo mediante simulación de escenarios realistas.

## 4.6 Recursos y Evaluación

- Cada caso estándar tiene un peso del **20%** en la nota final. - Los casos especiales pueden incrementar la nota **hasta un 20% adicional** sobre la calificación base. - Se proporcionarán datos predefinidos, incluyendo demandas y ubicaciones, para facilitar la comparación entre las soluciones. - Los datos para escenarios *single depot* y *multi-depot* estarán disponibles en el repositorio del proyecto, permitiendo a los estudiantes elegir su enfoque.

## 5 Entregables

Para esta etapa del proyecto, se requiere una serie de entregables que cubren desde la implementación del modelo hasta el análisis detallado de los resultados obtenidos en cada caso de prueba. El objetivo es asegurar que los estudiantes desarrollen una solución robusta y adaptable, capaz de manejar diferentes escenarios sin necesidad de ajustar el modelo para cada caso específico. A continuación, se detallan los entregables requeridos:

### 5.1 Implementación del Modelo

Se espera una versión actualizada del modelo matemático entregado en la fase anterior. Esta actualización debe incorporar los ajustes necesarios para soportar todos los casos de prueba de manera flexible. Los estudiantes deben documentar:

- Los cambios realizados al modelo para adaptarse a los distintos escenarios.
- La justificación de cada cambio y las ventajas identificadas al implementar estos ajustes.
- Asegurarse de que el modelo sea generalizable y no requiera modificaciones adicionales para cada caso de prueba.

El modelo debe implementarse en una clase en el lenguaje de programación elegido, con el código organizado para recibir los mismos formatos de archivos que los datos de los casos de prueba, facilitando así su evaluación. Cualquier intervención manual en los datos podría resultar en penalizaciones.

### 5.2 Rutas de los Vehículos

Para cada caso, los estudiantes deben entregar un archivo en formato **.csv** que documente las rutas seguidas por cada vehículo de manera clara y estructurada. Este archivo permitirá evaluar las secuencias de nodos visitados por cada vehículo y analizar la efectividad del modelo en el cumplimiento de las demandas logísticas.

**Formato del Nombre del Archivo** El nombre del archivo debe seguir el siguiente formato:

`<nombre-del-grupo>-caso-<tipo-de-caso>-<numero-del-caso>-ruta.csv`

donde:

- `<nombre-del-grupo>` es el nombre o identificador del equipo de estudiantes.
- `<tipo-de-caso>` indica si el caso es estándar o especial (**estandar** o **especial**).
- `<numero-del-caso>` es el número del caso correspondiente (por ejemplo, 1, 2, etc.).

Por ejemplo, un archivo de rutas para el primer caso estándar del equipo "Optimizadores" sería nombrado como:

`Optimizadores-caso-estandar-1-ruta.csv`

Mientras que un archivo para el primer caso especial sería:

`Optimizadores-caso-especial-1-ruta.csv`

**Formato del Contenido del Archivo** Cada archivo debe seguir el siguiente formato de columnas en el interior, organizado en formato `csv` para facilitar su lectura y análisis:

`ID-Vehiculo, ID-Origen, ID-Destino`

donde:

- `ID-Vehiculo` es el identificador único del vehículo que realiza la ruta.
- `ID-Origen` es el identificador del nodo de inicio de la ruta o de la parada actual.
- `ID-Destino` es el identificador del siguiente nodo al que se dirige el vehículo en su ruta.

Cada línea del archivo representa una transición de un nodo a otro para un vehículo específico, registrando toda la secuencia de paradas hasta la finalización de su ruta. Este formato permite analizar el trayecto detallado de cada vehículo de manera sencilla.

**Ejemplo de Contenido** A continuación, se muestra un ejemplo del contenido esperado dentro de un archivo de rutas para ilustrar el formato:

```
ID-Vehiculo,ID-Origen,ID-Destino
V01,N001,N002
V01,N002,N005
V01,N005,N003
V02,N003,N004
V02,N004,N006
V02,N006,N001
```

En este ejemplo:

- El vehículo `V01` comienza su ruta en el nodo `N001`, se desplaza al nodo `N002`, y así sucesivamente hasta completar su ruta.
- El vehículo `V02` sigue su propia secuencia de nodos comenzando en `N003` y terminando en `N001`.

### 5.3 Valor de la Función Objetivo

Para cada caso de prueba, se debe entregar un archivo con el valor de la función objetivo obtenida, que debe incluirse en un archivo con el prefijo `Caso_Objetivo` o `CasoEspecial_Objetivo` (por ejemplo, `Caso1_Objetivo.txt`). Este archivo debe contener:

- El valor total de la función objetivo para el caso.

### 5.4 Informe de Costos Operacionales

Se debe incluir un informe detallado de los costos operacionales que contribuyen al valor de la función objetivo. Este informe debe desglosar cada tipo de costo involucrado, tales como costos de distancia, tiempo de operación, mantenimiento diario, recarga o combustible, y cualquier otro costo relevante en el cálculo de la función objetivo.

### 5.5 Visualización de Rutas

Para cada caso, se debe generar una visualización gráfica que muestre las rutas seguidas por cada vehículo. La visualización debe permitir identificar el recorrido de cada vehículo, mostrando claramente los nodos de inicio y los destinos. Los detalles sobre los requisitos específicos de la visualización se amplían en la sección de calificación y bonos.

### 5.6 Análisis de Resultados

Se debe presentar un análisis exhaustivo de los resultados obtenidos en cada caso de prueba, incluyendo:

- Identificación de patrones o tendencias observadas en las rutas optimizadas.
- Problemas o desafíos encontrados en la modelación.
- Mejoras potenciales para optimizar los resultados y sugerencias para ajustes futuros.

### 5.7 Caso Especial: Escalabilidad

Para el caso especial de escalabilidad, se requiere un informe adicional que describa por qué la solución propuesta es escalable y qué limitaciones (o *drawbacks*) tiene en comparación con un enfoque tradicional. Este informe debe incluir un análisis comparativo y una evaluación de las ventajas y desventajas en términos de rendimiento y eficiencia.

### 5.8 Código del Modelo

El código del modelo debe organizarse en una clase, la cual debe estar diseñada para recibir los datos en el formato especificado para cada caso de prueba. Esto facilitará su uso y revisión, permitiendo a los evaluadores ejecutar el modelo sin modificaciones adicionales. El modelo debe cumplir con los siguientes requisitos:

- El modelo debe ser capaz de adaptarse a cualquier caso de prueba utilizando el mismo código base.

- Los datos deben ser procesados directamente sin intervención manual para asegurar la reproducibilidad.
- La clase debe estar bien documentada, con instrucciones claras sobre su ejecución y los parámetros de entrada necesarios.

## 6 Sistema de Calificación

Este sistema de calificación está diseñado para evaluar de manera justa y completa los entregables del proyecto de ruteo vehicular, asegurando que cada aspecto relevante del trabajo realizado por los estudiantes sea considerado. La estructura de la calificación también está pensada para incentivar la experimentación y el enfoque innovador, brindando oportunidades para lograr una nota superior a la máxima establecida. A continuación, se detalla la ponderación de cada entregable y la forma en que se evaluarán los casos estándar y especiales.

### 6.1 Ponderación de los Cambios en el Modelo

Los cambios implementados en el modelo matemático representan un 15% de la nota total del proyecto. Se evaluará la capacidad del modelo para adaptarse a todos los casos de prueba sin necesidad de modificaciones adicionales, así como la claridad de la justificación de los cambios realizados y las mejoras observadas en la eficiencia del modelo. La calidad de la implementación y la flexibilidad del código también serán aspectos importantes en esta evaluación.

### 6.2 Evaluación de los Casos Estándar

Cada caso estándar incluye entregables específicos que serán evaluados con las siguientes ponderaciones:

- **Archivo CSV de Rutas (40%):** Este archivo debe contener rutas válidas para cada vehículo. La precisión en la secuencia de nodos y la coherencia en las rutas planificadas serán los principales criterios de evaluación.
- **Valor de la Función Objetivo y Análisis de Costos (15%):** El valor de la función objetivo debe estar documentado junto con un análisis detallado de los costos operativos. Se evaluará la precisión de los cálculos y la claridad del desglose de costos.
- **Visualización de Rutas (25%):** La visualización debe representar de forma clara y efectiva las rutas seguidas por cada vehículo. Se considerará la legibilidad, el diseño y la creatividad en la presentación gráfica.
- **Análisis de Resultados (20%):** Este análisis debe detallar los patrones observados, problemas encontrados y posibles mejoras en el modelo. La profundidad y calidad del análisis serán los principales aspectos evaluados.



**Distribución de los Casos Estándar** Cada caso estándar tiene un nivel de complejidad creciente y será evaluado de acuerdo a la siguiente distribución porcentual:

- **Caso 1: Base** 35% de la nota total. Completar el caso base asegura una calificación mínima de 2.5 en el proyecto.
- **Caso 2:** 12.5% de la nota total.
- **Caso 3:** 12.5% de la nota total.
- **Caso 4:** 12.5% de la nota total.

La suma de estos casos estándar, junto con la evaluación de los cambios en el modelo (15%) y la calidad del código (5%), proporciona un puntaje máximo de 4.625 en el proyecto.

### 6.3 Evaluación de los Casos Especiales

El propósito de los casos especiales es fomentar la creatividad, la innovación y la experimentación en la resolución del problema de ruteo vehicular. Estos casos especiales ofrecen una oportunidad para que los estudiantes demuestren habilidades avanzadas y enfoques originales, permitiéndoles obtener una nota superior a la máxima establecida. Los casos especiales tienen las siguientes ponderaciones:

- **Caso Especial 1: Múltiples Productos** - 15% de la nota total.
- **Caso Especial 2: Nodos de Recarga** - 25% de la nota total.
- **Caso Especial 3: Escalabilidad** - 35% de la nota total.

Completar los casos especiales puede llevar la calificación del proyecto por encima de 5, reflejando el esfuerzo adicional y la capacidad de aplicar técnicas avanzadas en la resolución de problemas complejos.

### 6.4 Calidad del Código del Modelo

La calidad del código es un aspecto fundamental en la evaluación del proyecto y representa un 5% de la nota total. Se evaluará la organización del código, la claridad en la estructura de la clase que contiene el modelo y la facilidad para trabajar con archivos de entrada en el formato especificado. El código debe ser funcional y reproducible sin intervención manual en los datos, cumpliendo con el estándar de uso de clases para parametrización flexible.

### 6.5 Bonificaciones y Premios

Además de las calificaciones base, se otorgarán bonificaciones y premios adicionales para reconocer la creatividad, calidad y desempeño excepcional en la presentación de los resultados. Estos incentivos están diseñados para motivar a los estudiantes a explorar enfoques innovadores y destacarse en el proyecto:

- **Bonificación de 0.5 puntos:** Se otorgará una bonificación adicional de 0.5 puntos a las visualizaciones de rutas más creativas y mejor desarrolladas, reconociendo el esfuerzo y la calidad visual en la representación de los datos. Esta bonificación se sumará a la nota final del proyecto.
- **Premio a la Mejor Visualización:** La visualización de rutas que destaque por su claridad, innovación y atractivo visual será seleccionada para un premio especial, que corresponde a un incremento del 20% en la nota total del proyecto. Este premio incentiva el uso de técnicas visuales avanzadas para comunicar los resultados de manera efectiva y profesional.
- **Exención de la Entrega 3 del Proyecto:** Como reconocimiento al trabajo sobresaliente, el equipo de instructores (incluidos los dos profesores de la magistral y el profesor del laboratorio) seleccionará dos grupos para eximirlos de la tercera entrega del proyecto. Los grupos seleccionados deberán, en cambio, realizar una presentación en clase, ya sea para los estudiantes del curso o para la clase de Logística de Ingeniería Industrial. Esta exención es un reconocimiento a la excelencia y una oportunidad para compartir las mejores prácticas y resultados con una audiencia más amplia.