

Proyecto Etapa II **Implementación:**

Optimización en logística para la empresa LogistiCo

Curso: ISIS-3302
Departamento de Ingeniería de Sistemas
Universidad de los Andes

1 Introducción General

Este documento unifica los enunciados de **tres diferentes enunciados** para la realización del proyecto de optimización en logística, cada uno con un enfoque particular. En conjunto, los proyectos buscan mejorar la eficiencia operativa en distintos escenarios, considerando elementos comunes como la planificación de rutas, la asignación de vehículos y la minimización de costos, pero diferenciándose en aspectos específicos que se detallan a continuación.

2 Acerca de LogistiCo

LogistiCo es una empresa líder en soluciones logísticas, reconocida por su capacidad de innovar y adaptarse a entornos operativos complejos. Su misión es optimizar la distribución de mercancías y la entrega de insumos críticos mediante la aplicación de modelos matemáticos avanzados y tecnologías de punta. Con el fin de atender las distintas necesidades del mercado, LogistiCo se organiza en tres divisiones principales:

- **Urban Logistics:** Se enfoca en la distribución y gestión de mercancías dentro de áreas metropolitanas. Esta división aborda los desafíos inherentes a la alta densidad de tráfico, restricciones de espacio en entornos urbanos, el alto volumen de entregas.
- **Rural/Offshore Logistics:** Dedicada a la logística en zonas rurales y áreas de difícil acceso, esta división enfrenta retos relacionados con infraestructuras

limitadas, distancias largas y condiciones ambientales adversas, garantizando la entrega oportuna en comunidades remotas o aisladas dentro de una ventana de tiempo específica.

- **National Logistics:** Responsable de la coordinación del transporte de mercancías a nivel nacional, esta división integra redes de transporte, optimizando rutas que conectan puertos con destinos finales en todo el país, cumpliendo con diversas normativas y restricciones operativas.

3 Entregables

Para la entrega correspondiente a esta etapa, los estudiantes deberán presentar un documento en formato PDF o *Jupyter Notebook* que incluya el desarrollo completo de la implementación de los modelos previamente formulados utilizando Pyomo, acompañado de los archivos de código fuente. Los entregables se evaluarán conforme a los siguientes criterios:

1. Implementación del Modelo Matemático (10 %):

- Correcta traducción del modelo matemático a código en Pyomo, respetando fielmente la estructura formal desarrollada en la etapa anterior.
- Adecuada definición de conjuntos, parámetros, variables de decisión, función objetivo y restricciones.
- Manejo eficiente de la sintaxis y las características específicas de Pyomo.

2. Procesamiento de Datos (5 %):

- Correcta lectura e interpretación de los datos de entrada desde los archivos proporcionados en Bloque Neón.
- Preprocesamiento adecuado de la información para su utilización en el modelo.
- Cálculo correcto de parámetros derivados.

3. Resolución del Modelo para Casos de Estudio (60 %):

- Caso 1 (15 %): Implementación correcta del CVRP estándar común a los tres proyectos.

-
- Caso 2 (25 %): Resolución efectiva del caso con datos simplificados específicos de cada proyecto.
 - Caso 3(20 %):
 - Interpretación clara y rigurosa de las soluciones obtenidas.
 - Análisis de sensibilidad sobre parámetros clave del modelo.
 - Identificación de insights relevantes para la operación logística.

4. Análisis de Resultados y Visualización (15 %):

- Visualizaciones adecuadas para cada caso particular (rutas, flujos, niveles de carga, etc.).

5. Documentación y Calidad del Código (10 %):

- Código limpio, bien estructurado y adecuadamente comentado.
- Modularidad y reutilización de componentes.
- Manejo efectivo de excepciones y casos límite.
- Documentación clara que permita comprender la implementación.

4 Instrucciones Generales

Para completar satisfactoriamente esta segunda etapa del proyecto, los estudiantes deberán seguir estas instrucciones:

1. Organización de Repositorio:

- Crear una estructura clara de directorios que separe código, datos de entrada, resultados y documentación.
- Incluir un archivo README detallando la organización y los pasos para ejecutar el código.

2. Lectura de Datos:

- Desarrollar rutinas para importar los datos desde los archivos proporcionados en Bloque Neón.
- Implementar validaciones básicas para garantizar la integridad de los datos.

3. Implementación en Pyomo:

- Traducir el modelo matemático a código Pyomo siguiendo las mejores prácticas.
- Garantizar la correspondencia exacta entre la formulación matemática (etapa 1) y la implementación en código.

4. Ejecución y Análisis:

- Resolver el modelo para los tres casos de estudio propuestos.
- Documentar el proceso de solución, incluyendo tiempo de ejecución, métodos utilizados y convergencia.
- Analizar la sensibilidad del modelo ante cambios en parámetros críticos.

5. Visualización de Resultados:

- Desarrollar visualizaciones informativas y claras de las soluciones obtenidas.
- Para el Proyecto A: visualizar las rutas de vehículos y niveles de carga.
- Para el Proyecto B: representar el cumplimiento de ventanas de tiempo y la evolución de la carga.
- Para el Proyecto C: ilustrar los puntos de recarga, niveles de combustible y las restricciones viales y peajes que se cursaron.

6. Documentación:

- Elaborar un informe técnico que detalle la implementación, los resultados y el análisis realizado.
- Incluir capturas de las visualizaciones y tablas resumen de los hallazgos principales.

5 Proyecto A: Optimización en la Planeación de Transporte Vehicular Urbana Para LogistiCo

5.1 Objetivo

El objetivo principal de esta fase del proyecto es implementar en Pyomo el modelo de optimización previamente formulado para la asignación de inventario y planificación de rutas de transporte vehicular en Bogotá. La implementación debe permitir la minimización efectiva de los costos operativos y de transporte, garantizando entregas eficientes mientras se respetan las limitaciones operativas de LogistiCo.

5.2 Contexto

Esta fase se enfoca en la traducción del modelo matemático a código ejecutable que pueda ser utilizado en un entorno real de operaciones. La implementación debe capturar fielmente todas las complejidades del problema logístico urbano en Bogotá, incluyendo la gestión de múltiples centros de distribución con capacidades limitadas, la asignación óptima de vehículos con restricciones de autonomía, y la planificación de rutas eficientes en un entorno urbano caracterizado por alta congestión y restricciones de movilidad.

5.3 Datos del Problema

Los datos de entrada para los tres casos de estudio se encuentran disponibles en Bloque Neón con los nombres `Proyecto_Caso_Base`, `Proyecto_A_Caso2` y `Proyecto_A_Caso3`. **IMPORTANTE:** Leer cuidadosamente el archivo `README.md` incluido en cada carpeta. Este contiene la descripción detallada de los archivos obligatorios, sus formatos y el contexto de cada conjunto de datos.

5.4 Instrucciones

1. **Implementación del Modelo Base (Caso 1: CVRP Estándar):**
 - Implementar en Pyomo un modelo básico de CVRP con un único centro de distribución y vehículos homogéneos.
 - Validar que la solución obtenida es factible y cumple con las restricciones.
2. **Extensión a Múltiples Centros de Distribución (Caso 2):**

-
- Incorporar los tres centros de distribución con capacidades limitadas.
 - Permitir asignación de inventario desde cualquiera de los centros.
 - Se espera solamente comprobar validez y factibilidad de la solución obtenida.

3. Escenario Realista (Caso 3):

- Adaptar el modelo a un entorno más realista con entre 50 y 100 clientes.
- Incluir selección de flota heterogénea y restricciones urbanas avanzadas.
- Generar un análisis exhaustivo con enfoque empresarial.

4. Análisis de Sensibilidad y Reportes:

- Para todos los casos:
 - Crear mapas interactivos con Folium que muestren:
 - Rutas completas por vehículo
 - Centros de distribución como nodos de salida
- Exclusivo para Caso 3:
 - Explicar a detalle los cambios en la formulación matemática que tuvieron que realizar para poder resolver un ejercicio de mayor tamaño, cuáles son los *trade-offs* que tuvieron que realizar para poder resolverlo y si recomendarían solvers *mip/lp* para resolver estos problemas de este tamaño.
 - Estudiar el impacto de variaciones en:
 - Costos de combustible ($\pm 20\%$)
 - Capacidad de los centros de distribución
 - Demandas de los clientes
 - Reportes detallados por centro de distribución y por vehículo:
 - Costo total de operación (combustible, mantenimiento)
 - Distancia total recorrida
 - Tiempo total estimado de operación
 - Carga total entregada
 - Estadísticas globales:
 - Distancia, tiempo y carga promedio por vehículo
 - Desviaciones estándares y distribuciones de los datos

- Estadísticas de desempeño del algoritmo
- Conclusiones que respondan preguntas estratégicas:
 - *"¿Cuáles son los parámetros iniciales que más afectan la logística urbana?"*
 - *"¿Dónde se presentan los mayores cuellos de botella?"*
 - *"¿Qué mejoras recomendaría a LogistiCo?"*
- Identificar los parámetros que generan mayor impacto en el costo total y estructura de rutas.

5. Verificación de Soluciones:

- Para cada caso, se debe generar un archivo de verificación `verificacion_casoX.csv` (donde X corresponde al número del caso).
- Este archivo permitirá el rastreo de las rutas que cada vehículo tomó y la carga inicial para cumplir con las demandas.
- **IMPORTANTE:** La ausencia del archivo de verificación o su incorrecto formato resultará en una penalización significativa en la nota final.
- El formato del archivo debe seguir estrictamente la estructura definida a continuación.

6. Formato del Archivo de Verificación:

Listing 1: Ejemplo de archivo `verificacion_caso1.csv`

```

1 VehicleId,DepotId,InitialLoad,RouteSequence,ClientsServed,
  DemandsSatisfied,TotalDistance,TotalTime,FuelCost
2 VEH001,CDA,750,CDA-C005-C023-C017-CDA
  ,3,215-320-215,28.4,67.2,98500
3 VEH002,CDB,680,CDB-C002-C009-C014-CDB
  ,3,250-180-250,32.1,75.8,112300
4 VEH003,CDA,920,CDA-C001-C006-C012-C019-CDA
  ,4,300-220-150-250,41.5,112.3,145200

```

7. Explicación del Formato:

- **VehicleId:** Identificador único del vehículo, debe coincidir con el ID en el archivo de entrada.
- **DepotId:** Identificador del centro de distribución de origen, debe coincidir con los IDs en el archivo de entrada.

-
- **InitialLoad**: Carga inicial del vehículo al salir del centro de distribución (en unidades).
 - **RouteSequence**: Secuencia completa de la ruta incluyendo el depot inicial y final, separada por guiones.
 - **ClientsServed**: Número total de clientes atendidos en la ruta.
 - **DemandsSatisfied**: Lista de demandas satisfechas por cada cliente en el orden visitado, separadas por guiones.
 - **TotalDistance**: Distancia total recorrida (en km).
 - **TotalTime**: Tiempo total de la ruta (en minutos).
 - **FuelCost**: Costo total de combustible para la ruta (en pesos colombianos).

8. Consideraciones Adicionales:

- Para el Caso 2 y 3, se debe especificar correctamente el centro de distribución asignado a cada vehículo.
- Los valores numéricos deben utilizar punto como separador decimal.
- Las secuencias de valores deben separarse con guiones (-) sin espacios.
- El archivo debe tener encabezados exactamente como se muestra en el ejemplo.
- Se verificará que la suma de demandas satisfechas para cada cliente coincida con la demanda total requerida.
- Se validará que las cargas iniciales no excedan la capacidad de los vehículos especificada en los datos de entrada.

9. Entregables:

- Código Pyomo modular y comentado.
- Scripts para ejecutar cada caso.
- Archivos de verificación para cada caso (verificacion_caso1.csv, verificacion_caso2.csv, verificacion_caso3.csv).
- Visualizaciones y gráficos exportables (PDF o PNG).
- Documento de análisis en PDF.
- Carpeta con datos procesados y archivos README detallados.

6 Proyecto B: Optimización en la Planificación de Entregas Salva Vidas con Flota Híbrida para LogistiCo

6.1 Objetivo

El objetivo de esta fase es implementar en Pyomo el modelo de optimización para la planificación de entregas de insumos médicos y productos esenciales en comunidades remotas de La Guajira utilizando una flota híbrida de vehículos. La implementación debe permitir cumplir con las ventanas de tiempo establecidas minimizando costos operativos y maximizando el servicio a las comunidades.

6.2 Contexto

Esta implementación debe traducir a código ejecutable todas las complejidades asociadas con la operación en zonas remotas, incluyendo la gestión de ventanas de tiempo críticas para entregas médicas, la asignación **óptima entre drones y vehículos terrestres**, y la posibilidad de **reabastecimiento** para completar todas las entregas requeridas. El sistema debe ser lo suficientemente flexible para adaptarse a condiciones cambiantes y proporcionar planes de entrega viables en un entorno operativamente desafiante.

6.3 Datos del Problema

Los datos de entrada para los tres casos de estudio se encuentran disponibles en Bloque Neón bajo las etiquetas `Proyecto_Caso_Base`, `Proyecto_B_Caso2` y `Proyecto_B_Caso3`. Cada conjunto de datos incluye:

- Centro de distribución con coordenadas geográficas
- Clientes con demandas, ubicación y ventanas de tiempo
- Flota híbrida (vehículos terrestres y drones) con capacidades y autonomía
- Parámetros operativos adicionales como costos de operación y tiempos de recarga

6.4 Instrucciones

1. Caso Base (Caso 1: CVRP Estándar sin ventanas de tiempo):

- Implementar un CVRP básico con un centro de distribución y un solo tipo de vehículo.
- No se consideran ventanas de tiempo ni reabastecimiento.
- Validar factibilidad y cumplimiento de restricciones básicas.

2. Ventanas de Tiempo con Flota Híbrida (Caso 2):

- Ampliar el modelo para incluir drones y camionetas con capacidades distintas.
- Incorporar ventanas de tiempo como restricciones duras.
- No se permite reabastecimiento.

3. Escenario Realista con Reabastecimiento (Caso 3):

- Permitir múltiples viajes por vehículo, incluyendo retornos al depósito.
- Respetar ventanas de tiempo incluso con la lógica de reabastecimiento.
- Analizar secuencias temporales y su impacto en nivel de servicio.

4. Verificación de Soluciones:

- Para cada caso, se debe generar un archivo de verificación `verificacion_casoX.csv` (donde X corresponde al número del caso).
- Este archivo permitirá el rastreo de las rutas que cada vehículo tomó, la carga, los tiempos de llegada y los reabastecimientos.
- **IMPORTANTE:** La ausencia del archivo de verificación o su incorrecto formato resultará en una penalización significativa en la nota final.
- **NOTA:** Para el Caso 1 (base) no es necesario incluir información sobre tiempos de llegada ya que no se consideran ventanas de tiempo en este caso.

5. Formato del Archivo de Verificación:

Listing 2: Ejemplo de archivo `verificacion_caso2.csv` (con ventanas de tiempo)

```
1 VehicleId,VehicleType,InitialLoad,RouteSequence,ClientsServed,DemandSatisfied,  
  → ArrivalTimes,TotalDistance,TotalTime,Cost
```

```

2 | DRN001,Drone,35,CD-COM04-COM11-CD,2,15-20,10:35-11:15,28.5,55.0,185000
3 | TRK002,Truck,150,CD-COM02-COM07-COM09-CD
  | → ,3,45-60-45,09:45-10:30-11:15,42.8,120.0,275000
4 | DRN003,Drone,40,CD-COM01-COM05-CD,2,25-15,09:15-10:05,32.1,65.0,200000

```

Listing 3: Ejemplo de archivo verificacion_caso3.csv (con reabastecimientos)

```

1 | VehicleId,VehicleType,InitLoad,RouteSequence,Clients,DemandSatisfied,
  | → ArrivalTimes,Resup,ResupAmounts,Distance,Time,Cost
2 | DRN001,Drone,35,CD-COM04-COM11-CD-COM08-COM12-CD
  | → ,4,15-20-25-25,10:35-11:15-12:30-13:20,1,50,62.5,185.0,350000
3 | TRK002,Truck,150,CD-COM02-COM07-COM09-CD-COM15-COM21-CD
  | → ,5,45-60-45-75-70,09:45-10:30-11:15-13:00-14:15,1,145,78.3,250.0,425000
4 | DRN003,Drone,40,CD-COM01-COM05-CD-COM06-CD
  | → ,3,25-15-40,09:15-10:05-11:30,1,40,48.7,135.0,280000

```

6. Explicación del Formato:

- **VehicleId**: Identificador único del vehículo, debe coincidir con el ID en el archivo de entrada.
- **VehicleType**: Tipo de vehículo ("Drone." "Truck").
- **InitLoad/InitialLoad**: Carga inicial del vehículo al salir del centro de distribución (en unidades).
- **RouteSequence**: Secuencia completa de la ruta incluyendo el CD inicial, las comunidades y los retornos al CD.
- **Clients/ClientsServed**: Número total de clientes atendidos en la ruta.
- **DemandSatisfied**: Lista de demandas satisfechas por cada cliente en el orden visitado, separadas por guiones.
- **ArrivalTimes**: Hora de llegada a cada cliente en formato HH:MM, separadas por guiones.
- **Resup/Resupplies**: Número de veces que el vehículo retornó al CD para reabastecerse (Caso 3).
- **ResupAmounts**: Cantidad total reabastecida en cada retorno al CD, separadas por guiones (Caso 3).
- **Distance/TotalDistance**: Distancia total recorrida (en km).
- **Time/TotalTime**: Tiempo total de la ruta (en minutos).
- **Cost/OperationCost**: Costo total de operación para la ruta (en pesos colombianos).

7. Consideraciones Adicionales:

-
- Para el Caso 1, se puede omitir la columna de tiempos de llegada y reabastecimientos.
 - Para el Caso 2, se debe omitir las columnas de reabastecimientos pero incluir la de tiempos de llegada.
 - Para el Caso 3, todas las columnas son obligatorias.
 - Los valores numéricos deben utilizar punto como separador decimal.
 - Las secuencias de valores deben separarse con guiones (-) sin espacios.
 - El centro de distribución debe identificarse como ÇD.^{en} la secuencia de ruta.
 - Las comunidades o clientes deben identificarse con el formato ÇOMXX"donde XX es el número del cliente.
 - El formato de hora debe ser HH:MM (24 horas).
 - Se verificará que todos los clientes sean atendidos dentro de sus ventanas de tiempo especificadas.
 - Se validará que las cargas iniciales y reabastecimientos no excedan la capacidad de los vehículos.

8. Consideraciones Adicionales:

- Para el Caso 1, se puede omitir las columnas de ventanas de tiempo y reabastecimientos.
- Para el Caso 2, se debe omitir las columnas de reabastecimientos pero incluir las de ventanas de tiempo.
- Para el Caso 3, todas las columnas son obligatorias.
- Los valores numéricos deben utilizar punto como separador decimal.
- Las secuencias de valores deben separarse con guiones (-) sin espacios.
- El centro de distribución debe identificarse como ÇD.^{en} la secuencia de ruta.
- Las comunidades o clientes deben identificarse con el formato ÇOMXX"donde XX es el número del cliente.
- El formato de hora debe ser HH:MM (24 horas).
- Se verificará que todos los clientes sean atendidos dentro de sus ventanas de tiempo especificadas.

-
- Se validará que las cargas iniciales y reabastecimientos no excedan la capacidad de los vehículos.

9. Visualización y Reportes:

- Para todos los casos:
 - Mapas de rutas diferenciadas por tipo de vehículo (dron o terrestre)
 - Diagramas de Gantt con cumplimiento de ventanas de tiempo por cliente (excepto Caso 1)
- Exclusivo para Caso 3:
 - Gráficos y tablas por vehículo:
 - Tiempo de llegada a cada cliente
 - Cantidad de entregas realizadas
 - Número de reabastecimientos
 - Costo total, distancia y tiempo de operación
 - Análisis de sensibilidad para:
 - Variaciones en tiempos de recarga o vuelo
 - Cambios en demanda
 - Fallo temporal de un tipo de vehículo
 - Conclusiones que respondan a:
 - *"¿Se cumplieron las ventanas de tiempo en promedio?"*
 - *"¿Cuál fue el número promedio de reabastecimientos por tipo de vehículo?"*
 - *"¿Qué configuración es más robusta para zonas remotas?"*

10. Entregables:

- Código modular Pyomo y scripts de ejecución.
- Archivos de verificación para cada caso (verificacion_caso1.csv, verificacion_caso2.csv, verificacion_caso3.csv).
- Visualizaciones: mapas, Gantt, gráficos comparativos.
- Documento de análisis en PDF con interpretaciones clave.
- Carpeta de datos con archivo README explicativo.

7 Proyecto C: Optimización en la Planificación de Rutas Nacionales y Estrategia de Recarga para LogistiCo

7.1 Objetivo

El objetivo de esta fase es implementar en Pyomo el modelo previamente formulado para la planificación de rutas nacionales de transporte de carga con estrategias **óptimas de recarga de combustible**. La implementación debe minimizar los costos totales considerando distancias, peajes, recargas estratégicas y restricciones de peso por municipio.

7.2 Contexto

La implementación debe traducir a código ejecutable todas las complejidades del transporte nacional en Colombia, incluyendo la variación de precios de combustible entre estaciones, el impacto de los peajes en los costos operativos, y las restricciones de peso impuestas por diferentes municipios. El sistema debe optimizar no solo las rutas sino también las decisiones estratégicas sobre dónde y cuánto recargar combustible para cumplir con las restricciones de autonomía y carga.

7.3 Datos del Problema

Los datos para los tres casos de estudio se encuentran disponibles en Bloque Neón bajo las etiquetas `Proyecto_Caso_Base`, `Proyecto_C_Caso2` y `Proyecto_C_Caso3`. Cada conjunto incluye:

- Punto de acceso (Puerto de Barranquilla) con sus coordenadas
- Centros de consumo (municipios) con demandas y restricciones de peso máximo
- Estaciones de servicio con ubicaciones y precios de combustible
- Peajes con sus tarifas base y variables según el peso
- Características de la flota de camiones (capacidad, autonomía, consumo)

7.4 Instrucciones

1. Caso Base (Caso 1: CVRP Estándar):

- Implementar un modelo básico tipo CVRP con un **origen nacional** (*puerto*) y destinos (**municipios**).
- Incluir restricciones de capacidad y autonomía de los vehículos.
- Validar factibilidad de la solución considerando solamente distancia y demanda.

2. Caso Intermedio: Incorporación de Recarga (Caso 2):

- Extender el modelo anterior para incluir decisiones de recarga.
- Tomar en cuenta los diferentes precios de combustible en estaciones a lo largo del recorrido.
- Asegurar que ningún vehículo se quede sin combustible en ninguna parte de la ruta.
- Este caso permite probar estrategias como recarga completa vs. recarga mínima necesaria.

3. Escenario Complejo con Peajes y Restricciones de Peso (Caso 3):

- Incorporar restricciones de peso por municipio, asociadas a normativas locales.
- Incluir peajes con tarifas variables según el peso y tramo recorrido.
- Determinar estrategias **conjuntas de ruteo y recarga**, optimizando el costo total nacional.

4. Formato del Archivo de Verificación:

Listing 4: Ejemplo de archivo verificacion_caso1.csv (básico)

```
1 VehicleId,LoadCap,FuelCap,RouteSequence,Municipalities,DemandSatisfied,
  ↳ InitialLoad,InitialFuel,Distance,Time,TotalCost
2 CAM001,20000,200,PT0-MUN03-MUN07-MUN12-PT0
  ↳ ,3,5200-7800-6500,19500,200,310.5,270.2,580000
3 CAM002,15000,150,PT0-MUN02-MUN05-MUN09-PT0
  ↳ ,3,4500-5200-4800,14500,150,285.7,250.5,530000
4 CAM003,25000,250,PT0-MUN01-MUN06-MUN11-MUN15-PT0
  ↳ ,4,6000-5500-6800-6200,24500,250,420.3,380.7,750000
```

Listing 5: Ejemplo de archivo verificacion_caso2.csv (con recarga)

```

1 VehicleId,LoadCap,FuelCap,RouteSequence,Municipalities,DemandSatisfied,
  ↳ InitLoad,InitFuel,RefuelStops,RefuelAmounts,Distance,Time,FuelCost,
  ↳ TotalCost
2 CAM001,20000,200,PTO-MUN03-EST02-MUN07-MUN12-PTO
  ↳ ,3,5200-0-7800-6500,19500,180,1,150,380.5,320.2,175000,680000
3 CAM002,15000,150,PTO-MUN02-EST01-MUN05-MUN09-PTO
  ↳ ,3,4500-0-5200-4800,14500,120,1,100,325.7,290.5,142000,590000
4 CAM003,25000,250,PTO-MUN01-EST03-MUN06-MUN11-EST04-MUN15-PTO
  ↳ ,4,6000-0-5500-6800-0-6200,24500,200,2,170-120,530.3,460.7,230000,890000
  ↳

```

Listing 6: Ejemplo de archivo verificacion_caso3.csv (completo)

```

1 VehicleId,LoadCap,FuelCap,RouteSeq,Municipalities,Demand,InitLoad,InitFuel,
  ↳ RefuelStops,RefuelAmounts,TollsVisited,TollCosts,VehicleWeights,
  ↳ Distance,Time,FuelCost,TollCost,TotalCost
2 CAM001,20000,200,PTO-MUN03-EST02-PEA01-MUN07-PEA03-MUN12-PTO,3
  ↳ ,5200-0-0-7800-0-6500,19500,180,1,150,2,85000-92000
  ↳ ,19500-14300-6500,420.5,360.2,175000,177000,980000
3 CAM002,15000,150,PTO-MUN02-EST01-PEA02-MUN05-PEA04-MUN09-PTO,3
  ↳ ,4500-0-0-5200-0-4800,14500,120,1,100,2,70000-65000
  ↳ ,14500-10000-4800,350.7,325.5,142000,135000,790000
4 CAM003,25000,250,PTO-MUN01-EST03-PEA01-MUN06-PEA03-MUN11-EST04-PEA05-MUN15-
  ↳ PTO,4,6000-0-0-5500-0-6800-0-0-6200,24500,200,2,170-120,3
  ↳ ,95000-92000-105000,24500-18500-13000-6200,580.3,520.7,230000
  ↳ ,292000,1150000

```

5. Explicación del Formato:

- **VehicleId:** Identificador único del vehículo.
- **LoadCap:** Capacidad de carga del vehículo (en kg).
- **FuelCap:** Capacidad de combustible del vehículo (en galones).
- **RouteSeq/RouteSequence:** Secuencia completa de la ruta incluyendo puerto (PTO), municipios (MUN), estaciones de servicio (EST) y peajes (PEA).
- **Municipalities:** Número de municipios (clientes) atendidos en la ruta.
- **Demand/DemandSatisfied:** Lista de demandas satisfechas en cada punto de la ruta (0 para estaciones y peajes).
- **InitLoad/InitialLoad:** Carga inicial del vehículo al salir del puerto (en kg).
- **InitFuel/InitialFuel:** Combustible inicial del vehículo al salir del puerto (en galones).
- **RefuelStops:** Número de paradas para recargar combustible (Casos 2 y 3).

-
- **RefuelAmounts:** Cantidades de combustible recargadas en cada parada (en galones).
 - **TollsVisited:** Número de peajes por los que pasa el vehículo (solo Caso 3).
 - **TollCosts:** Costo en cada peaje visitado (en pesos colombianos).
 - **VehicleWeights:** Peso del vehículo al entrar a cada municipio (en kg) (solo Caso 3).
 - **Distance:** Distancia total recorrida (en km).
 - **Time:** Tiempo total de la ruta (en minutos).
 - **FuelCost:** Costo total de combustible (en pesos colombianos).
 - **TollCost:** Costo total de peajes (solo Caso 3).
 - **TotalCost:** Costo total de la operación (en pesos colombianos).

6. Consideraciones Adicionales:

- Para el Caso 1, se pueden omitir las columnas relacionadas con recargas y peajes.
- Para el Caso 2, se deben incluir las columnas de recargas pero se pueden omitir las de peajes.
- Para el Caso 3, todas las columnas son obligatorias.
- Los valores numéricos deben utilizar punto como separador decimal.
- Las secuencias de valores deben separarse con guiones (-) sin espacios.
- El puerto de origen debe identificarse como "PTO.^{en} la secuencia de ruta.
- Los municipios deben identificarse con el formato "MUNXX" donde XX es el número del municipio.
- Las estaciones de servicio deben identificarse como ".^{ES}TXX" donde XX es el número de la estación.
- Los peajes deben identificarse como "PEAXX" donde XX es el número del peaje.
- Se verificará que las cargas no excedan la capacidad de los vehículos.
- Se validará que los vehículos no se queden sin combustible en ningún tramo.

-
- Se comprobará que los vehículos respeten las restricciones de peso por municipio especificadas en los datos de entrada (Caso 3).

7. Visualización y Reportes:

- Para todos los casos:
 - Mapas que muestren las rutas nacionales planificadas
 - Visualización básica de los puntos de recarga (cuando aplique)
- Exclusivo para Caso 3:
 - Mapas detallados que muestren:
 - Rutas nacionales con paradas de recarga específicas
 - Municipios con restricciones y su cumplimiento
 - Tablas por vehículo que reporten:
 - Cantidad total recargada y puntos de recarga
 - Costo por componente: combustible, peajes, mantenimiento
 - Tiempo y distancia totales por ruta
 - Peso transportado vs permitido por municipio
 - Análisis de sensibilidad ante:
 - Aumento de precios en estaciones clave (+20 %)
 - Reducción de autonomía por deterioro de camiones
 - Inclusión o exclusión de estaciones estratégicas
 - Conclusiones que respondan a:
 - *"¿Dónde debería LogistiCo establecer acuerdos con estaciones para minimizar costos?"*
 - *"¿Qué tipo de camiones son más eficientes según el patrón de demanda?"*
 - *"¿Cómo afectan los peajes variables la asignación óptima de rutas?"*

8. Entregables:

- Código Pyomo con estructura modular.
- Scripts de ejecución para cada caso.
- Archivos de verificación para cada caso (verificacion_caso1.csv, verificacion_caso2.csv, verificacion_caso3.csv).

-
- Visualizaciones interactivas (folium) y estáticas (matplotlib).
 - Documento PDF con análisis completo, recomendaciones y tablas resumen.
 - Carpeta de datos con README detallado.

Consideraciones finales para todos los proyectos:

- Todos los modelos deben estar implementados utilizando Pyomo para permitir su solución con solvers comerciales o de código abierto.
- El código debe estar modularizado para facilitar su mantenimiento y extensión.
- Se debe incluir manejo adecuado de errores y validación de datos de entrada.
- Las visualizaciones deben ser informativas y orientadas a facilitar la toma de decisiones operativas.
- La documentación debe ser suficientemente clara para permitir que otros desarrolladores puedan entender y modificar el código.

Tener en cuenta:

- No se reciben entregas por fuera del plazo máximo y tampoco por correo. Las entregas solo se reciben por **Bloque Neón**.
- Esta actividad se puede entregar en grupos de hasta 3 integrantes.
- **IMPORTANTE!!!** Junto a los archivos de la entrega adjuntar los nombres de los integrantes del equipo.