



---

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

FACULTAD DE INGENIERÍA

MAESTRÍA EN CIENCIAS DE LA INGENIERÍA

TESIS

MODELO DE OPTIMIZACIÓN DE RUTAS PARA EL SERVICIO DE  
TRANSPORTE UNIVERSITARIO

Para obtener el grado de:  
Maestro en Ciencias de la Ingeniería

Presenta:  
Ing. en Comp. Marco Antonio Sánchez Carmona

Director de tesis:  
Dr. José Raymundo Marcial Romero

Tutores:  
Dra. Lourdes Loza Hernández  
Dr. José Antonio Hernández Servín

Toluca, Edo. de México

Diciembre de 2025

*“Bueno es que dejes ya esa desidia tuya, pues no es entre almohadones de plumas ni bajo techo como se logra la fama, y cuando se vive sin haberla cultivado, se deja un rastro similar al del humo en el aire o la espuma sobre el agua. Así, pues, levántate y trata de vencer tu deidad, procurando crear en ti el ánimo del que triunfa en los combates, de quien no se deja dominar por el cuerpo cansado pues es mucho el camino que hay que recorrer y no se limita a este infierno; aprovecha mis palabras, si es que bien las comprendes”.*

*(Dante Alighieri, La divina comedia, Infierno, Canto XXIV, versos 52–55)*

## Resumen

La optimización aplicada al área de transporte puede abordarse desde diversas perspectivas. Entre las más comunes se encuentran la determinación de rutas y las mejoras orientadas a la reducción de costos, tales como el traslado de pasajeros, emisiones de carbono, los costos de viaje, entre otros. El transporte de estudiantes es una de las áreas donde se ha observado que se requiere optimizar, debido al aumento continuo y la dispersión geográfica de la población estudiantil, lo cual repercute en problemas de traslado por la cantidad limitada de vehículos (autobuses) disponibles.

La presente tesis aborda el análisis del caso de estudio del servicio de transporte de la Universidad Autónoma del Estado de México. Se propone el uso de diversos procesos de clustering, utilizando el algoritmo de Agrupamiento Espacial Basado en Densidad de Aplicaciones con Ruido, en conjunto con tres métricas que permiten realizar una evaluación de calidad en los clusters formados, con el fin de decretar si los puntos establecidos como destinos y paradas son adecuados para los alumnos. Adicionalmente, se implementa una envolvente convexa para la identificación del subconjunto de puntos terminales desde los cuales deben iniciar las rutas de transporte.

Con los conjuntos de datos definidos a partir de los procesos anteriores, se establecieron los parámetros y datos indispensables para plantear un modelo de Programación Lineal Entera, cuyo objetivo es generar un nuevo conjunto de rutas de transporte adecuadas para cada uno de los vehículos disponibles. El modelo considera restricciones retomadas del conocido Problema de Enrutamiento de Vehículos e incorpora variables de decisión y restricciones adicionales para movilizar la mayor cantidad posible de alumnos hacia sus destinos mediante la formulación del Problema de Flujo Máximo. Como resultado, se diseñó una función multi-objetivo orientada a encontrar rutas de costo mínimo, con flujo máximo de alumnos por autobús, integrando además tiempos de salida y espera.

Las pruebas del modelo se realizaron mediante tres escenarios con variaciones en la cantidad de nodos, destinos, terminales y vehículos. La solución obtenida con el solver COIN-OR Branch and Cut, permitió generar rutas óptimas para cada autobús en los escenarios 1 y 2, logrando trasladar la mayor cantidad de alumnos a sus destinos correspondientes a un costo mínimo (tiempo), y asegurando que las rutas inicien y finalicen en horarios adecuados con respecto a los horarios de entrada de los estudiantes. Sin embargo, para el escenario 3, el tiempo requerido por el solver para encontrar una solución óptima fue considerablemente mayor.



# Índice general

---

Índice general	v
Índice de tablas	vii
Índice de figuras	ix
<b>1. Introducción</b>	<b>1</b>
1.1. Definición del objeto de estudio . . . . .	2
1.2. Justificación . . . . .	3
1.3. Meta de ingeniería . . . . .	4
1.4. Objetivos . . . . .	4
1.4.1. Objetivo general . . . . .	4
1.4.2. Objetivos específicos . . . . .	4
1.5. Metodología . . . . .	5
<b>2. Marco teórico</b>	<b>7</b>
2.1. Clustering . . . . .	7
2.1.1. Agrupamiento Espacial Basado en Densidad de Aplicaciones con Ruido	7
2.1.2. Métricas para la determinación de la distancia . . . . .	10
2.1.3. Métricas de evaluación para algoritmos de clustering . . . . .	11
2.1.4. Envolvente Convexa . . . . .	14
2.1.5. Estimación de Densidad de Kernel . . . . .	15
2.2. Optimización y problemas de optimización . . . . .	15
2.3. Investigación de operaciones . . . . .	15
2.4. Problema de Enrutamiento de Vehículos . . . . .	16
2.5. Flujo en redes . . . . .	20
2.5.1. Problema de Flujo Máximo . . . . .	21
2.6. Principios para una programación y diseño de rutas . . . . .	22
2.7. Métodos de solución comunes para el Problema de Enrutamiento de Vehículos . . . . .	23

2.7.1. Métodos exactos . . . . .	23
2.7.2. Métodos aproximados . . . . .	25
<b>3. Estado del arte</b>	<b>29</b>
<b>4. Marco metodológico</b>	<b>37</b>
4.1. Estudio cuantitativo . . . . .	38
4.2. Procedimientos de análisis de datos . . . . .	38
4.3. Aplicación de procesos de clustering . . . . .	38
4.4. Modelo matemático . . . . .	39
4.5. Solución del modelo con métodos exactos y aproximados . . . . .	39
4.6. Implementación de un prototipo de sistema . . . . .	39
<b>5. Análisis de resultados</b>	<b>41</b>
5.1. Preprocesamiento de los datos . . . . .	41
5.1.1. Descripción de los conjuntos de datos y su función . . . . .	42
5.1.2. Preprocesamiento de la matrícula de alumnos . . . . .	42
5.2. Proceso de clustering . . . . .	46
5.2.1. Clustering de Espacios Académicos . . . . .	46
5.2.2. Proceso de clustering de la matrícula de alumnos . . . . .	52
5.2.3. Proceso de clustering de la matrícula de alumnos con distinción de horarios . . . . .	55
5.2.4. Análisis por capas en un horario determinado . . . . .	57
5.2.5. Definición de los datos de salida para “alimentar” el modelo matemático de ILP . . . . .	58
5.3. Modelos de ILP para el caso de estudio . . . . .	62
5.3.1. Consideraciones generales para los modelos de ILP . . . . .	63
5.3.2. Modelo a dos índices . . . . .	63
5.3.3. Modelo de VRP con flujo máximo . . . . .	65
5.4. Visualización de las rutas obtenidas por los modelos . . . . .	71
5.4.1. Resultados del modelo de CVRP a dos índices . . . . .	71
5.4.2. Resultados del modelo de IDA a flujo máximo . . . . .	74
5.5. Comparativa de las rutas obtenidas y las rutas actuales . . . . .	80
5.6. Prototipo de sistema . . . . .	82
<b>6. Conclusiones y trabajo futuro</b>	<b>89</b>
6.1. Conclusiones . . . . .	89
6.2. Trabajo Futuro . . . . .	91
<b>Referencias</b>	<b>93</b>

# Índice de tablas

---

3.1. Resumen de la información de los artículos consultados (Elaboración propia).	33
5.1. Horarios de entrada propuestos (Elaboración propia).	42
5.2. Distribución de la totalidad de alumnos por EA (Elaboración propia).	43
5.3. Clusters formados de los Espacios Académicos (Elaboración propia).	47
5.4. Resultados de las métricas de valoración de calidad de clusters (Elaboración propia).	50
5.5. Resultados de las métricas de valoración de calidad de clusters con normalización(Elaboración propia).	51
5.6. Resultados de los tiempos de ejecución de las tres etapas de DBSCAN (Elaboración propia).	60
5.7. Valoración de las rutas obtenidas del modelo a dos índices (Elaboración propia).	72
5.8. Configuraciones de los escenarios de pruebas (Elaboración propia).	74
5.9. Resultados de las pruebas con los distintos escenarios (Elaboración propia).	78
5.10. Resultados de las pruebas con los distintos escenarios (Elaboración propia).	78
5.11. Rutas obtenidas del escenario 1 y su valoración (Elaboración propia).	79
5.12. Rutas obtenidas del escenario 2 y su valoración (Elaboración propia).	79
5.13. Distancias de las rutas actuales para el sentido de IDA (Elaboración propia).	81



# Índice de figuras

---

1.1. Diagrama de bloques de la metodología para el desarrollo del trabajo (Elaboración propia). . . . .	5
2.1. Comparativa del TSP y VRP (Elaboración propia). . . . .	17
2.2. Gráfica no dirigida con cuatro vértices y cinco aristas (Elaboración propia). . . . .	18
2.3. Gráfica dirigida con cuatro vértices y cinco aristas (Elaboración propia). . . . .	18
2.4. Ejemplo de un VRPTW, cada cliente tienen una ventana de tiempo $[a,b]$ (Elaboración propia). . . . .	20
2.5. Ejemplo de red dirigida para el problema de flujo máximo (Elaboración propia). . . . .	22
4.1. Diagrama de bloques de la metodología (Elaboración propia). . . . .	37
5.1. Distribución de los alumnos para cada EA dentro de la zona de servicio (Elaboración propia). . . . .	45
5.2. Proceso de clustering de los EA (Elaboración propia). . . . .	48
5.3. Conteo de clusters y outliers al variar el parámetro EPS (Elaboración propia). . . . .	52
5.4. Resultados del proceso de clustering en un plano de dos dimensiones (Elaboración propia). . . . .	53
5.5. Proceso de clustering con tres características (Elaboración propia). . . . .	54
5.6. Volumen de alumnos en dos horarios distintos (Elaboración propia). . . . .	55
5.7. Proceso de clustering con tres características en un horario de 7:00 am (Elaboración propia). . . . .	56
5.8. Análisis de la capa 3 - Campus CU (Elaboración propia). . . . .	57
5.9. Análisis mediante KDE para la capa 3 (Elaboración propia). . . . .	58
5.10. Puntos determinados por los procesos de clustering (Elaboración propia). . . . .	59
5.11. Implementación de la Envoltura Convexa (Elaboración propia). . . . .	61
5.12. Puntos finales para el modelo ILP (Elaboración propia). . . . .	62
5.13. Rutas obtenidas con un modelo de ILP de dos índices (Elaboración propia). . . . .	72

5.14. Rutas actuales y puntos de parada del proceso de clustering (Elaboración propia). . . . .	73
5.15. Rutas de IDA obtenidas en un horario de 7:00 am para los escenarios 1 y 2 (Elaboración propia). . . . .	75
5.16. Muestra de las rutas de IDA obtenidas en un horario de 7:00 am para el escenario 1 (Elaboración propia). . . . .	76
5.17. Muestra de las rutas de IDA obtenidas en un horario de 7:00 am para el escenario 2 (Elaboración propia). . . . .	77
5.18. Muestra de las rutas de IDA actuales (Elaboración propia). . . . .	80
5.19. Diagrama de Casos de Uso para el prototipo de sistema (Elaboración propia). . . . .	82
5.20. Interfaz general del prototipo de sistema (Elaboración propia). . . . .	83
5.21. Interfaz para las rutas actuales (Elaboración propia). . . . .	84
5.22. Proceso de clustering (Elaboración propia). . . . .	85
5.23. Prototipo de sistema (Elaboración propia). . . . .	86
5.24. Configuraciones para el clustering (Elaboración propia). . . . .	87
5.25. Configuraciones para el solver (Elaboración propia). . . . .	88

---

# Capítulo 1

## Introducción

---

El transporte urbano es fundamental para realizar nuestras actividades cotidianas, así como para el desarrollo adecuado de las ciudades, ya que se mejora los tiempos y eficiencia del traslado de los pasajeros [1, 2]. Incluso, algunos trabajos afirman que “en la mayoría de las áreas metropolitanas se suele dar mayor importancia a los medios de transporte que tienden a cubrir un mayor volumen de pasajeros como el transporte ferroviario”[1].

La importancia del transporte también lo argumentan instituciones de renombre como el Banco Mundial en el año 2023, que hace la siguiente afirmación: “el transporte es fundamental para respaldar el crecimiento económico, crear empleo y conectar a las personas con los servicios esenciales; como la atención a la salud o a la educación”[2]. Se hace mención de que actualmente, hay cerca de 100 millones de personas cuya vivienda está a más de 2 kilómetros de un camino que sea transitable y la falta de este acceso está relacionada a la carencia de recursos económicos [2]. Otro aspecto fundamental son los problemas ambientales, ya que es prioritario reducir el impacto climático, debido a que el transporte aporta un incremento en las emisiones globales de Gases de Efecto Invernadero (GEI) [3, 4].

Por otro lado, en el ámbito de la logística, el transporte es un área de decisión clave y de suma importancia, aunque los elementos sobre el transporte se representan de diferentes maneras, los principales son: la selección del tipo de transporte (aéreo, marítimo o terrestre), el diseño de las rutas, la programación (horarios) de los vehículos y los acuerdos para el envío (para transporte de bienes) [5]. La mayoría de los costos, relacionados al transporte, se encuentran entre un tercio y dos tercios de los costos totales para el área logística [5], lo que implica que el mejorar la eficiencia, con el máximo uso del equipo de transporte y del personal, es de atención primordial [5].

Gracias al desarrollo de herramientas algorítmicas y de modelado, que consideran las características de los problemas de distribución que han surgido del mundo real, se han encontrado *soluciones* en tiempos computacionales aceptables [6]. Generalmente los problemas concernientes a la distribución de bienes entre depósitos y usuarios finales

(clientes) son conocidos como Problema de Enrutamiento de Vehículos (VRP, por sus siglas en inglés) [6]. En el modelado de este tipo de problemas, se tiene el objetivo de optimizar algún recurso (costos operativos); por ello es de importancia mencionar el área de la *optimización*, abordada más adelante en los siguientes capítulos, así como la Investigación de Operaciones (IO) para la correcta toma de decisiones.

La presente propuesta de trabajo de tesis, consiste en el planteamiento de un modelo matemático de Programación Lineal Entera (ILP, por sus siglas en inglés), tomando de referencia el VRP y sus diversas categorías, así como el problema conocido como flujo en redes; todo esto para representar la mayor cantidad de características observables y cuantificables (cantidad de vehículos, capacidades, afluencia diaria, distribución de los recursos, entre otros) del servicio de transporte de la Universidad Autónoma del Estado de México (UAEMex), tomado como caso de estudio, para ello se deben establecer un conjunto de restricciones para las capacidades de los vehículos, demanda de los usuarios, tiempo de servicio y ventanas de tiempo (en caso de ser necesario) para las paradas de autobús; para finalmente resolverlo con un método adecuado a la complejidad del modelo y generar nuevas rutas de transporte que serán comparadas con las rutas actuales. Los datos necesarios para “alimentar” el modelo son obtenidos principalmente de la matrícula de alumnos de la UAEMex, los cuales son procesados por un algoritmo de clustering y una envolvente convexa, con el objetivo de determinar puntos de parada adecuados para el servicio de transporte propuesto. Todos estos componentes serán puestos en un prototipo de sistema que utilice principalmente los datos de la matrícula de alumnos inscritos, los espacios académicos (EA) de la UAEMex, la definición de una zona de servicio, donde opera el transporte, y finalmente mostrar la solución del modelo mediante rutas sobre un mapa.

## 1.1. Definición del objeto de estudio

La UAEMex cuenta con un servicio de transporte para uso exclusivo de los estudiantes llamado *potrobús*, el cual permite a los alumnos trasladarse desde ciertas zonas hasta los principales espacios académicos de la universidad ubicados en el Valle de Toluca. El potrobús opera a través de 16 rutas de transporte, tanto de **ida** y **regreso**. Para cada ruta, existe un número determinado de autobuses que la recorren (mínimo una unidad); de la totalidad de rutas existentes algunas no tienen convergencia en el campus principal llamado C.U., es decir, que existen otros puntos donde las rutas tienen lugares de salida y destino como son: la Terminal de autobuses de Toluca y el edificio de Rectoría de la UAEMex. Cada una de las rutas contiene un número determinado de paradas ubicadas en tres categorías: Ascenso, Descenso y Ascenso-Descenso.

Con la información proporcionada por la dirección de transporte de la UAEMex, se han determinado los siguientes aspectos:

- El servicio se satura de lunes a viernes, principalmente en los horarios de: 6:00 AM, 1:00 PM, 3:00 PM y 6:00 PM.

- El servicio en otros horarios cubre menos del 30 % de su capacidad de asientos.
- Los autobuses llevan más del doble de pasajeros de la cantidad de asientos que tienen, llegando incluso a llevar el triple de los mismos; esto es notorio para las rutas a Ixtlahuca y Santiago Tianguistenco en horarios de 6:00 am y 6:00 pm.
- Hay paradas de autobús en zonas poco convenientes para el abordaje de personas.
- En algunas rutas hay puntos de parada ubicados muy cerca uno de otros.
- Los puntos de parada no cuentan con ventanas de tiempo ni tiempo de servicio para una espera adecuada del vehículo.
- En ciertos horarios no se utiliza toda la flota de vehículos.

Estos aspectos, permiten realizar el siguiente planteamiento de problema:

La UAEMex, en su servicio de transporte institucional, requiere implementar un modelo de optimización de rutas basado en el VRP o alguna de sus variaciones, combinando un proceso de clustering para la determinación de puntos de parada adecuados, considerando el incremento de la matrícula de alumnos, así como, los cambios periódicos relacionados a la ubicación de estos, todo esto para generar un conjunto de rutas de transporte óptimas. La optimización permitirá conocer los recursos necesarios para que el servicio no esté saturado y sea suficiente en ciertos horarios.

## 1.2. Justificación

El estudio aporta un modelo matemático que considera la mayor cantidad de los elementos que describen las particularidades de un “servicio de transporte universitario”: número de paradas de autobús distribuidas desde el campus principal hasta municipios aledaños, distribución de las residencias de los alumnos, ubicación de los EA, restricciones de capacidad y tiempo para los vehículos y rutas respectivamente. Los datos de la matrícula de alumnos son usados en un proceso de clustering para determinar los puntos adecuados para establecer paradas de autobús. Además, el método de solución adecuado al modelo se adapta a las necesidades que se tienen hoy en día para dar una respuesta a las problemáticas que se viven continuamente en el área de la dirección de transporte de la UAEMex.

Se puede conseguir un impacto en un ámbito “Social y ambiental” debido a que, analizando los resultados, se podrá reducir rutas o autobuses innecesarios, así como puntos de parada redundantes y canalizar recursos (unidades de autobuses) en rutas donde se requiera una mayor demanda. De este modo se optimizan tiempos tanto de los usuarios como de trayecto en cuanto a distancias recorridas y a su vez, puede ser reflejado en un ahorro de combustible.

El estudio puede generar nuevas metodologías para la optimización de rutas de transporte tanto en servicios públicos y privados, sin embargo para este estudio se centró únicamente en el transporte institucional con el que cuenta la UAEMex.

### **1.3. Meta de ingeniería**

El uso de un proceso de clustering basado en densidad, aunado a la implementación de un modelo matemático de programación lineal entera, para el servicio del potrobús, mejora el conjunto de rutas de transporte, considerando cantidad y capacidades de los vehículos, así como la demanda y horarios de los usuarios.

### **1.4. Objetivos**

#### **1.4.1. Objetivo general**

Crear un modelo de optimización de rutas a través de un proceso de clustering y programación lineal entera para mejorar el servicio de transporte de la UAEMex, considerando los recursos disponibles.

#### **1.4.2. Objetivos específicos**

Los objetivos específicos que son propuestos para llevar a cabo este proyecto se mencionan en el siguiente listado:

1. Analizar el estado actual del servicio del potrobús, mediante datos oficiales y la observación del mismo para tener un panorama de su funcionamiento.
2. Determinar la demanda del servicio de transporte institucional a través de un instrumento estadístico (encuesta) y recolección de datos de afluencia por ruta y horarios para establecer los datos a usarse en el modelo matemático.
3. Utilizar un proceso de clustering en la matrícula de alumnos para determinar puntos de parada de autobús y utilizarlos en el modelo matemático.
4. Diseñar un modelo matemático con programación lineal entera para representar las características del servicio de transporte (capacidad de los vehículos, demanda del servicio, horarios, tipo de vehículo, ventanas de tiempo)
5. Solucionar el modelo diseñado a través de la aplicación de una herramienta de optimización adecuada para obtener las nuevas rutas de transporte.
6. Comparar las nuevas rutas obtenidas con las rutas actuales utilizando la métrica definida en la función objetivo del modelo, para definir la mejora en las rutas de transporte.

7. Construir un prototipo de sistema que permita la generación de las rutas con los componentes realizados previamente, para ser usado por la dirección de transporte.

## 1.5. Metodología

La línea de trabajo establecida para el desarrollo de esta investigación, tomando como base lo propuesto por Liu y Wang [1], se muestra mediante el diagrama de la Figura 1.1:

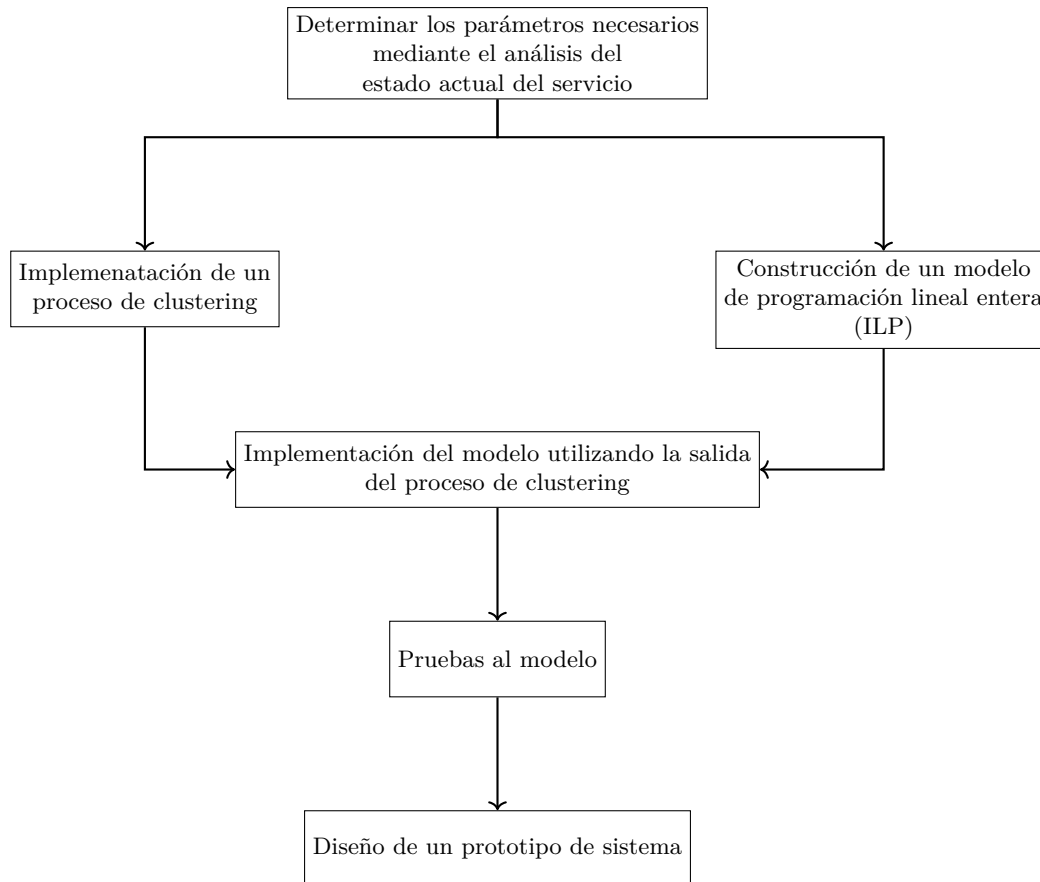


Figura 1.1: Diagrama de bloques de la metodología para el desarrollo del trabajo (Elaboración propia).

La descripción completa de cada una de las etapas de la metodología propuesta es:

1. Determinar los parámetros necesarios del caso de estudio, mediante el análisis de los informes de afluencia diaria proporcionados por la Dirección de Transporte de la UAEMex, así como la aplicación de encuestas para determinar: número de rutas, cantidad de autobuses, características de los autobuses, afluencia de alumnos que utilizan el servicio y horarios de mayor demanda.
2. Realizar un proceso de clustering con los datos de la matrícula de alumnos considerando: espacios académicos, códigos postales (de la residencia de los alumnos), horarios y la zona de servicio del transporte (Valle de Toluca y municipios aledaños).
3. Construcción de un modelo de ILP, retomando características de la literatura sobre VRP y Flujo Máximo, estableciendo una función multi-objetivo y un conjunto de restricciones adecuadas al caso de estudio aunado a los datos obtenidos y revisados previamente.
4. Implementación del modelo utilizando los grupos formados por el proceso de clustering para la formación de nuevas rutas de transporte.
5. Elaboración de pruebas al modelo para determinar la mejora en comparación a las rutas actuales del servicio.
6. Diseño de un prototipo de sistema en línea, que permita el uso de los elementos desarrollados anteriormente para los usuarios y tomadores de decisiones de la Dirección de Transporte.

---

# Capítulo 2

## Marco teórico

---

Los siguientes elementos teóricos mostrados son mencionados y descritos con mayor profundidad en lo realizado por Sánchez et al. en [7, 8], por lo que se retoma lo esencial para este trabajo.

### 2.1. Clustering

El término clustering es un proceso que consiste en formar grupos, también denominados clusters, con elementos que presentan características similares. Se utiliza principalmente para determinar patrones climáticos, análisis cartográficos, análisis de datos, segmentación de imágenes, agrupación de artículos por temas o para segmentar clientes, entre otros [7, 9]. Es conocido como una técnica del área del *Machine Learning*, cuya finalidad es separar datos para conformar grupos *homogéneos*, es decir, que presenten características comunes [10, 11].

#### 2.1.1. Agrupamiento Espacial Basado en Densidad de Aplicaciones con Ruido

Es un algoritmo para el agrupamiento de datos, se conoce simplemente como DBSCAN (Density-Based Spatial Clustering of Applications with Noise), fue planteado por Martin Ester et al. [12]. Se le denomina y clasifica como un algoritmo de agrupamiento basado en densidad, porque permite hallar un número de grupos iniciando con una estimación en la distribución de densidad de los puntos apropiados [12]. Las características y funcionamiento de dicho algoritmo son retomados de lo mencionado por [7], de las cuales cabe destacar las siguientes:

## Características y funcionamiento

Las características de DBSCAN que permite su eficiencia en el agrupamiento de datos se debe a:

- En cada observación se inspecciona el número de puntos a una distancia máxima denotada como  $Eps$ , esta zona se denomina  $Eps$ -vecindad de la observación.
- Si al menos en una observación tiene cierto número de vecinos, incluida ella misma, se considera una observación central.
- Todas las observaciones realizadas en la vecindad de una observación central pertenecen al mismo grupo.
- Cualquier observación que no sea una observación central y que no tenga ninguna observación central en su vecindad se considera una *anomalía* [12].

## Algoritmo

El procedimiento general que establece el algoritmo DBSCAN, retomado de [7, 12], es el siguiente:

1. Encontrar los puntos en la  $Eps$ -vecindad de cada punto, e identificar los puntos centrales con la mayor cantidad de  $minPts$  vecinos.
2. Encontrar los componentes conectados de los puntos centrales en la gráfica vecina, ignorando todos los puntos no centrales.
3. Asignar cada punto no central a un grupo cercano si el grupo es un vecino  $Eps$ ; de lo contrario, asignarlo como un dato no etiquetado comúnmente denominado ruido (en este trabajo se le denomina *outlier*).

El proceso descrito se continúa hasta tener un grupo *densamente* conectado, algún punto que no haya sido visitado se analiza y se vuelve a procesar con la finalidad de descubrir otro grupo o algún elemento outlier [12].

El algoritmo original DBSCAN [12], se compone de tres elementos: algoritmo principal, ver Algoritmo 1, la función *ExpandCluster*, ver Algoritmo 2 y el método *regionQuery*, ver Algoritmo 3.

---

**Algorithm 1** DBSCAN

---

**Ensure:** *setOfPoints, eps, minPts*

$n \leftarrow \text{size}(\text{setOfPoints})$

$\text{clusterId} \leftarrow \text{UNCLASSIFIED}$

$\text{visited}[n] \leftarrow \text{FALSE}$

$\text{clusterAssignment} \leftarrow \text{nelementos, inicializada0}$

**for**  $i \leftarrow 1$  TO  $n$  **do**

**if**  $\text{visited}[i] \leftarrow \text{TRUE}$  **then**

$\text{continue}$

**end if**

$\text{visited}[i] = \text{TRUE}$

$\text{neighbors} = \text{regionQuery}(\text{setOfPoints}, i, \text{eps})$

**if**  $\text{size}(\text{neighbors}) < \text{minPts}$  **then**

$\text{clusterAssignment}[i] \leftarrow \text{NOISE}$

**else**

$\text{clusterId} \leftarrow \text{clusterId} + 1$

$\text{ExpandCluster}(\text{setOfPoints}, i, \text{neighbors}, \text{clusterId}, \text{eps}, \text{minPts},$   
       $\text{visited}, \text{clusterAssignment})$

**end if**

**end for**

**return**  $\text{clusterAssignment}$

---

---

**Algorithm 2** ExpandCluster

---

**Require:** *setOfPoints, point, neighbors, clusterId, eps, minPts, visited,*  
   $\text{clusterAssignment}$

$\text{clusterAssignment}[\text{point}] \leftarrow \text{clusterId}$

$i \leftarrow 1$

**while**  $i \leftarrow \text{size}(\text{neighbors})$  **do**

$p \leftarrow \text{neighbors}[i]$

**if**  $\text{!visited}[p]$  **then**

$\text{visited}[p] \leftarrow \text{TRUE}$

$\text{newNeighbors} \leftarrow \text{regionQuery}(\text{setOfPoints}, p, \text{eps})$

**if**  $\text{size}(\text{newNeighbors}) \geq \text{minPts}$  **then**

$\text{append}(\text{neighbors}, \text{newNeighbors})$

**end if**

**end if**

**if**  $\text{clusterAssignment}[p] == 0$  **then**

$\text{clusterAssignment}[p] \leftarrow \text{clusterId}$

**end if**

$i = i + 1$

**end while**

---

---

**Algorithm 3** RegionQuery

---

```
Require: setOfPoints, eps  
neighbors[]  
n ← size(setOfPoints)  
for j ← 1 TO n do  
  lat1 ← setOfPoints[i][1]  
  lon1 ← setOfPoints[i][2]  
  lat2 ← setOfPoints[j][1]  
  lon2 ← setOfPoints[j][2]  
  if distanceHaversine(lat1, lon1, lat2, lon2) ≤ eps then  
    neighbors.push(j)  
  end if  
end for  
return neighbors
```

---

## Complejidad

El algoritmo visita cada punto en todo el conjunto de puntos y posiblemente lo realiza varias veces con el fin de determinar otros elementos que puedan ser candidatos a pertenecer a otro clusters. En la práctica, DBSCAN presenta una complejidad temporal que se debe principalmente al número de solicitudes del método denominado *regionQuery* [13]. Es importante aclarar que DBSCAN ejecuta solamente una consulta por cada uno de los puntos, y si se utiliza una estructura de tipo índice, la complejidad temporal total sería de  $O(n \log n)$ ; sin usar la estructura mencionada, la complejidad temporal sería  $O(n^2)$  [13]. Con una matriz de distancias de tamaño  $(n^2 - n)/2$  se evita recalcular las distancias, lo que implica una complejidad  $O(n^2)$  de memoria, por otro lado, con una implementación que no utilice matrices solo necesitaría  $O(n)$  de memoria [12, 13].

### 2.1.2. Métricas para la determinación de la distancia

Los algoritmos de clustering basados en densidad necesitan de un parámetro de distancia, por lo que es de importancia definir adecuadamente una métrica para este componente y obtener una cantidad de clusters adecuada [7, 14], en DBSCAN es más notorio esta situación debido al parámetro *Eps*, otros algoritmos del mismo tipo que DBSCAN en la comparativa realizada en [7] determinan dicha distancia “automáticamente”. La distancia puede ser medida bajo una métrica en particular dependiendo de la naturaleza de los datos. Las principales métricas de distancia utilizadas son:

#### Distancia Euclidiana

Es un número positivo que muestra la separación presente entre dos puntos en un espacio donde se cumplen los axiomas y teoremas de la geometría de Euclides [14]. En un espacio euclidiano la distancia entre dos puntos, expresados como  $p = (x_1, y_1)$  y  $q = (x_2, y_2)$ , es

simplemente la longitud del vector  $pq$  que pertenece a la única recta que pasa por los puntos  $p$  y  $q$ . Para un espacio de dos dimensiones, la fórmula de la distancia euclidiana se muestra en la Ecuación 2.1.

$$d(p,q) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2.1)$$

En un espacio de  $n$ -dimensiones, la distancia Euclidiana existente entre los puntos conformados por:  $P = (p_1, p_2, \dots, p_n)$  y  $Q = (q_1, q_2, \dots, q_n)$ , se puede definir mediante la Ecuación 2.2.

$$d_E(P,Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (2.2)$$

### Fórmula Haversine

Es una expresión matemática cuyo uso sirve para calcular la distancia entre dos puntos sobre una esfera mediante sus coordenadas en formato: longitud y latitud [15]. En pocas palabras, permite calcular la distancia más corta entre dos puntos sobre la superficie de un objeto esférico, como la Tierra [15]. La fórmula de Haversine, ver Ecuación 2.3, se suele utilizar para el cálculo de rutas de vuelo, distancias en mapas e incluso la navegación marítima [16]. Otro de sus usos es en la meteorología, ya que permite calcular las trayectorias de las tormentas, posicionar ciertos equipos de telecomunicaciones y para trazar la trayectoria de los satélites [15, 17].

$$d = 2 \cdot R \cdot \arcsin \sqrt{\sin^2 \left( \frac{\varphi_2 - \varphi_1}{2} \right) + \cos \varphi_1 \cdot \cos \varphi_2 \cdot \sin^2 \left( \frac{\lambda_2 - \lambda_1}{2} \right)} \quad (2.3)$$

donde:

- $\varphi_1, \varphi_2$  son las latitudes del punto 1 y 2 respectivamente.
- $\lambda_1, \lambda_2$  son las longitudes del punto 1 y 2 respectivamente.
- $R$  representa el radio de la esfera (siendo el radio de la Tierra, que equivale a 6,371 km).

La implementación de la fórmula Haversine no puede ser usada para más de dos dimensiones, esto ocasiona que en procesos de clustering con más de dos características no pueda ser aplicada como métrica de distancia, en esas situaciones es ideal utilizar la distancia Euclidiana como métrica [7].

### 2.1.3. Métricas de evaluación para algoritmos de clustering

Para determinar la calidad de los clusters obtenidos por cualquier algoritmo de clustering, es necesario el uso de métricas de evaluación [7]. Las métricas de valoración para este ámbito, se pueden clasificar de manera general en: *Métricas internas* y *Métricas externas* [18].

Las métricas internas permiten evaluar la calidad de los clusters usando solamente los datos y las etiquetas sin necesidad de conocer las clases verdaderas; por otro lado las métricas externas miden qué tan bien el algoritmo de clustering reproduce las categorías verdaderas [18]. Algunas de las métricas internas más utilizadas son [19, 20]:

### Silhouette Score

Mide la similaridad de los puntos que se ubican dentro de un cluster y se aplica una comparación con los puntos dentro de otros clusters, su valor oscila entre  $-1$  y  $1$ , donde  $1$  indica clusters bien definidos [21]. El proceso del cálculo de Silhouette Score es el siguiente:

1. Se calcula la distancia promedio, expresada como  $a(i)$ , entre el punto  $i$  y todos los demás puntos que están en el mismo cluster mediante la Ecuación 2.4, siendo  $C_i$  el conjunto de puntos y  $\|x_i - x_j\|$  la distancia entre los puntos  $i$  y  $j$ .

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, j \neq i} \|x_i - x_j\| \quad (2.4)$$

2. Se calcula otra distancia promedio, expresada como  $b(i)$ , entre el punto  $i$  y todos los otros puntos presentes en el cluster más cercano (que no sea el mismo cluster  $i$ ) mediante la Ecuación 2.5, donde  $C_k$  es un cluster diferente al cluster  $C_i$ .

$$b(i) = \min_{k \neq C_i} \frac{1}{|C_k|} \sum_{j \in C_k} \|x_i - x_j\| \quad (2.5)$$

3. Se realiza el cálculo del índice de Silhouette para el punto  $i$  con la Ecuación 2.6.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (2.6)$$

4. Finalmente se calcula el Silhouette score para el conjunto completo con la Ecuación 2.7, siendo esta el promedio del índice de Silhouette de todos los puntos, donde  $N$  representa la totalidad de puntos que están en el conjunto de datos.

$$S = \frac{1}{N} \sum_{i=1}^N s(i) \quad (2.7)$$

### Índice Davies-Bouldin

Evalúa la dispersión de los clusters y la distancia que está presente entre ellos, un valor más bajo indica clusters mejor formados [22]. El procedimiento para obtener el índice Davies-Bouldin es el siguiente:

1. Se obtienen la dispersión de cada cluster utilizando la Ecuación 2.8, siendo  $C_i$  el conjunto de puntos,  $c_i$  el centroide del cluster,  $\|x - c_i\|$  la distancia existente entre el punto  $x$  y el centroide, y  $|C_i|$  la cantidad de puntos en el cluster, todo esto para el punto  $i$ .

$$S_i = \frac{1}{|C_i|} \sum_{x \in C_i} \|x - c_i\| \quad (2.8)$$

2. El cálculo de la separación entre clusters se consigue aplicando la Ecuación 2.9, donde  $c_i$  y  $c_j$  son los centroides de los clusters  $i$  y  $j$  respectivamente; y siendo  $\|c_i - c_j\|$  la distancia presente entre los centroides de dichos clusters.

$$M_{ij} = \|c_i - c_j\| \quad (2.9)$$

3. La medida de Davies-Bouldin para cada par de clusters se obtienen con la Ecuación 2.10.

$$R_{ij} = \frac{S_i + S_j}{M_{ij}} \quad (2.10)$$

4. Se calcula el máximo de  $R_{ij}$  para cluster  $i$  sobre todos los clusters  $j \neq i$  utilizando la Ecuación 2.11.

$$D_i = \max_{j \neq i} R_{ij} \quad (2.11)$$

5. Finalmente se obtienen el índice de Davies-Bouldin con la Ecuación 2.12, donde  $N$  es el número total de clusters.

$$DBI = \frac{1}{N} \sum_{i=1}^N D_i \quad (2.12)$$

### Índice Calinski-Harabasz

Se le conoce como el criterio de varianza de la relación, el cual evalúa la dispersión existente entre clusters y la cohesión dentro de los clusters. Un valor más alto indica clusters mejor formados [23].

1. Se obtiene la matriz de dispersión entre clusters ( $B_k$ ) mediante la Ecuación 2.13, siendo  $k$  el número de clusters,  $C_i$  el conjunto de puntos dentro del cluster  $i$ ,  $|C_i|$  la cantidad de puntos del cluster  $i$ ,  $c_i$  el centroide del cluster  $i$  y finalmente  $c$  es el centroide global de todos los datos.

$$B_k = \sum_{i=1}^k |C_i| (c_i - c)(c_i - c)^T \quad (2.13)$$

2. Se calcula la matriz de dispersión dentro de clusters  $W_k$  con la Ecuación 2.14

$$W_k = \sum_{i=1}^k \sum_{x \in C_i} (x - c_i)(x - c_i)^T \quad (2.14)$$

3. El cálculo del índice de Calinski-Harabasz ( $CH$ ) [23], se obtienen finalmente aplicando la Ecuación 2.15, donde  $trace(B_k)$  es la traza de la matriz de dispersión entre clusters,  $trace(W_k)$  es la traza de la matriz de dispersión dentro de clusters y  $n$  siendo el número total de puntos del conjunto de datos.

$$CH = \frac{trace(B_k)/(k-1)}{trace(W_k)/(n-k)} \quad (2.15)$$

En este trabajo se emplean solamente las tres métricas mencionadas debido a su capacidad para medir simultáneamente la separación entre clusters y la cohesión interna. Aunque el *Índice Dunn* también es una métrica interna que se ha reportada en la literatura, su uso no es necesario en este estudio, ya que tiende a ser más sensible a valores atípicos y su cálculo se vuelve costoso computacionalmente en conjuntos grandes, además de que su implementación no es adecuada para evaluar clusters basados en densidad como DBSCAN [24, 25].

#### 2.1.4. Envolverte Convexa

La envolvente convexa (convex hull por su nombre en inglés) de un conjunto de puntos, es el polígono convexo más pequeño que contiene a todos los puntos de dicho conjunto [8, 26]. En términos geométricos, es como *envolver* un conjunto de puntos en una “membrana elástica” que se ajusta alrededor de ellos, de tal forma que cualquier línea que conecte dos puntos dentro de la envolvente también se encuentra dentro de esta [8, 26, 27]. Existen varios algoritmos comunes para calcular el convex hull [8], entre ellos:

- Algoritmo de Graham (Graham scan). Ordena los puntos por el ángulo polar respecto a un punto base y, usando un recorrido de lista, identifica los puntos que pertenecen a la envolvente convexa.
- Algoritmo de Jarvis (Gift wrapping). Selecciona puntos sucesivos de la envolvente convexa hasta cerrar el polígono.
- Algoritmo de Quickhull: Utiliza la estrategia “Divide y conquista” (utilizada en algoritmos de ordenación), para encontrar los puntos extremos en el espacio.

En la Ecuación 2.16 se muestra la expresión de la envolvente convexa  $C$  para un conjunto de puntos  $X$ , donde dado  $k$  puntos  $x_1, x_2, \dots, x_k$  [8, 26]. La envolvente convexa se define como:

$$C(X) = \left\{ \sum_{i=1}^k \alpha_i x_i \mid x_i \in X, \alpha_i \in \mathbb{R}, \alpha_i \geq 0, \sum_{i=1}^k \alpha_i = 1 \right\} \quad (2.16)$$

### 2.1.5. Estimación de Densidad de Kernel

La Estimación de Densidad de Kernel (KDE, por sus siglas en inglés) es una técnica no paramétrica que es utilizada para *estimar* la función de densidad de probabilidad de una variable aleatoria [28]. KDE es útil para suavizar la distribución de datos y visualizar su estructura subyacente sin asumir una forma específica (como la normal) [28]. Se emplea un kernel (función de suavizado) que se centra en cada punto de datos, sumando los efectos de cada kernel para obtener una estimación suave de la densidad. Los elementos clave en KDE incluyen:

- Kernel. Una función que especifica la forma de los “bultos” sobre cada punto de datos, como la Gaussiana, Epanechnikov y uniforme.
- Ancho de banda. Parámetro que determina el ancho del kernel (un ancho de banda más pequeño produce estimaciones más detalladas pero propensas a ruido, mientras que uno más amplio genera una distribución más suave).

En la Ecuación 2.17 se puede observar la expresión para la determinación del KDE; siendo  $x_1, x_2, \dots, x_n$  muestras independientes e idénticamente distribuidas,  $x$  cualquier punto dado,  $K()$  es la función de Kernel, la cual es una función no negativa y  $h$  es un parámetro de suavizado [28], de tal manera que  $h > 0$ .

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^N K\left(\frac{x - x_i}{h}\right) \quad (2.17)$$

## 2.2. Optimización y problemas de optimización

La *optimización* es conocida como una herramienta importante en la toma de decisiones y en el análisis de sistemas físicos [29]. Para implementarla se deben identificar algunos objetivos y medidas cuantitativas del rendimiento del sistema bajo estudio. El *objetivo* puede ser beneficio y tiempo, además dicho objetivo también depende de ciertas características del sistema, llamadas *variables*. La meta es encontrar el valor de las variables que optimizan el objetivo [29].

Un *problema de optimización* radica en encontrar la *mejor solución* en un conjunto determinado de soluciones posibles, es decir, consiste en elegir el mejor elemento, con respecto a alguna base establecida, de un conjunto de elementos que se encuentran disponibles (tiende a buscar *máximos* y *mínimos*) [29].

## 2.3. Investigación de operaciones

La investigación de operaciones (IO) es definida como una disciplina que se especializa en la aplicación de métodos analíticos (numéricos) avanzados para facilitar la toma de mejores

decisiones [30]. La Sociedad Mexicana de Investigación de Operaciones (SMIO), la define como una subdisciplina de las matemáticas que usa métodos analíticos avanzados para ayudar en la toma de decisiones [31].

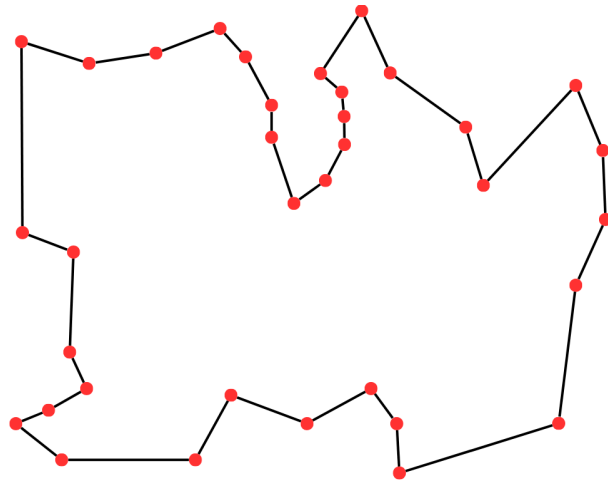
La IO utiliza técnicas de modelado matemático (determinista ó estocástico), análisis estadístico, simulación probabilista y optimización matemática para modelar, analizar y resolver problemas complejos que impiden la toma de decisiones [31]. La principal importancia de la IO es debido a su utilización para modelar problemas reales y facilitando la extracción de características de un problema para ser resuelto mediante algún método en particular [30].

El modelo general de IO se enfoca en maximizar o minimizar una *Función objetivo*, que se encuentra sujeta a un conjunto de *Restricciones*. Para que la solución de un modelo sea factible, ésta debe satisfacer todas las restricciones impuestas y es óptima si además genera el mejor valor (máximo o mínimo) de la función objetivo establecida [30].

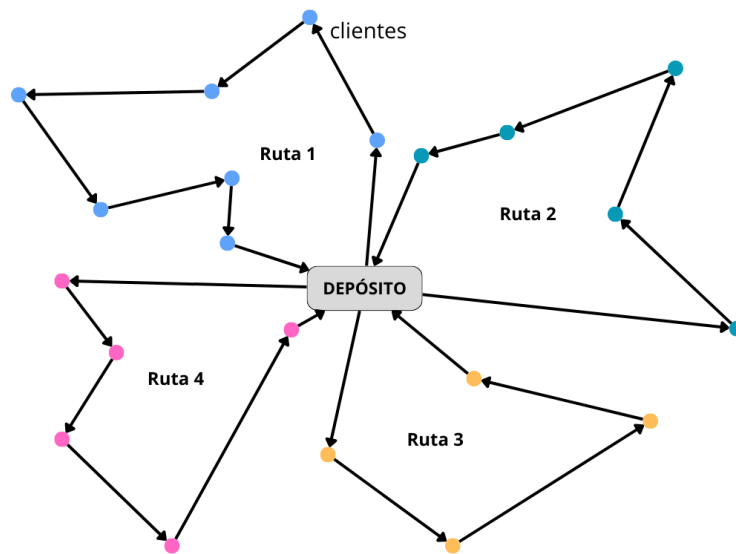
## 2.4. Problema de Enrutamiento de Vehículos

El VRP tiene sus orígenes del conocido Problema del Agente Viajero (TSP, por sus siglas en inglés), el cual parte de la idea donde un vendedor quiere visitar un conjunto de ciudades y regresar al punto de origen. Hay un costo de viaje desde una ciudad  $i$  a una ciudad  $j$  [6, 32]. El objetivo es visitar cada ciudad una sola vez, minimizando el costo total del viaje [33]. La representación del TSP se suele realizar mediante una gráfica, ver Figura 2.1a, y cada vértice en la gráfica representa una ciudad. Todos los vértices están conectados unos a otros y para cada camino (arista) existe un costo (distancia por lo regular). Conforme se tenga una gran cantidad de vértices, encontrar la mejor solución en ese conjunto de caminos tiende a ser complicado por la cantidad de permutaciones que se generan ( $n!$ ). El TSP solo considera una restricción la cual establece que solo se puede pasar por un vértice una única vez.

En [6] se hace mención del denominado Problema del Agente Viajero Múltiple (mTSP), donde la formulación del problema implica tener varios vendedores que parten de un mismo punto, cada vendedor debe visitar distintas ciudades, sin que ningún vendedor visite la misma ciudad, pero siguiendo el ideal del problema, es decir, que se visiten todas las ciudades minimizando el costo de ir de la ciudad  $i$  a  $j$  y finalmente todos los  $m$  vendedores deben retornar a la ciudad de partida, por lo cual ahora se generan  $m$  rutas como solución al problema. Partiendo del TSP y del mTSP, el VRP se enfoca en la determinación del conjunto óptimo de rutas que un grupo de vehículos debe realizar para un conjunto de clientes, ver Figura 2.1b. Es uno de los problemas combinatorios más importantes, ya que su importancia e interés se debe a su relevancia práctica, así como por su dificultad [6]. La primera mención y formulación del VRP aparece en el trabajo de George Dantzig y John Ramser en 1959, en donde se plantea una aproximación algorítmica cuyo fin es la entrega de gasolina [34].



(a) Ejemplo de solución para un problema de TSP.



(b) Ejemplo de solución para un VRP con un sólo depósito.

Figura 2.1: Comparativa del TSP y VRP (Elaboración propia).

Debido a que tanto el TSP y el VRP suelen ser representados mediante gráficas, por este motivo es relevante considerar los siguientes conceptos relacionados a la teoría de gráficas retomados de [35]:

**Gráfica.** Las gráficas son una composición de un conjunto de objetos conocidos como vértices (nodos) que se relacionan con otros vértices mediante un conjunto de caminos conocidos como aristas, matemáticamente se puede representar como  $G = (V, A)$ , donde  $V$



**Matriz de adyacencia.** Es una matriz numérica cuadrada que se utiliza para representar las relaciones binarias entre los vértices [35]. Las características más relevantes de este tipo de matriz son:

- Las columnas y filas representan los vértices de la gráfica.
- Por cada una de las aristas que unen a dos vértices, se suma un 1 al valor que en esa ubicación de la matriz.
- Si una arista es un bucle (loop) y la gráfica es no dirigida, entonces se le suma un 2.
- En caso de que la gráfica sea ponderada, entonces en lugar de sumar un 1 se suma el peso de la arista correspondiente.

Un ejemplo de matriz de adyacencia para la gráfica de la Figura 2.3, se muestra en 2.19:

$$\begin{array}{c}
 v_1 \\
 v_2 \\
 v_3 \\
 v_4
 \end{array}
 \begin{array}{cccc}
 v_1 & v_2 & v_3 & v_4 \\
 \left[ \begin{array}{cccc}
 0 & 1 & 1 & 0 \\
 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 1 \\
 1 & 0 & 0 & 0
 \end{array} \right]
 \end{array}
 \tag{2.19}$$

Una de las variaciones más notables de VRP utilizado principalmente para el transporte de carga, es aquel que considera horarios de abordaje y un tiempo de servicio, con ello surge el Problema de Enrutamiento de Vehículos con Ventanas de Tiempo (VRPTW, por sus siglas en inglés) [6]. Si se agrega una ventana de tiempo (TW) a cada cliente (vértice), además de la restricción de capacidad, un vehículo ahora tiene que visitar a un cliente dentro de un período de tiempo determinado, es decir, debe llegar en un tiempo  $a$  y salir en un tiempo  $b$ , denotado como  $[a, b]$ . En la Figura 2.4 se muestra una representación de un VRPTW con las TW en cada vértice [36]. El vehículo puede llegar antes de que se abra dicha ventana, pero no se puede atender al cliente hasta que se abran las ventanas de tiempo y no está permitido llegar después de que se haya cerrado la ventana. Existen más categorías de VRP que incluso parten de la categoría misma del VRPTW, como el Problema de Enrutamiento de Autobuses Escolares (SBRP, por sus siglas en inglés) [37], que sigue manteniendo la formulación y principios del VRPTW, pero considerando elementos y restricciones relacionadas a un conjunto de escuelas, tiempos de *timbre escolar*, consideraciones para la distribución de alumnos, entre otros aspectos. En el ámbito del transporte de pasajeros, es importante el modelaje de la recolección y entrega de dichos pasajeros a sus respectivos puntos de *subida* y *bajada* sin romper alguna restricción basada en la demanda y capacidad, por lo cual es importante considerar la categoría de VRP denominada Problema de Enrutamiento de Vehículos con Recolección y Entrega (VRPPD, por sus siglas en inglés) [6].

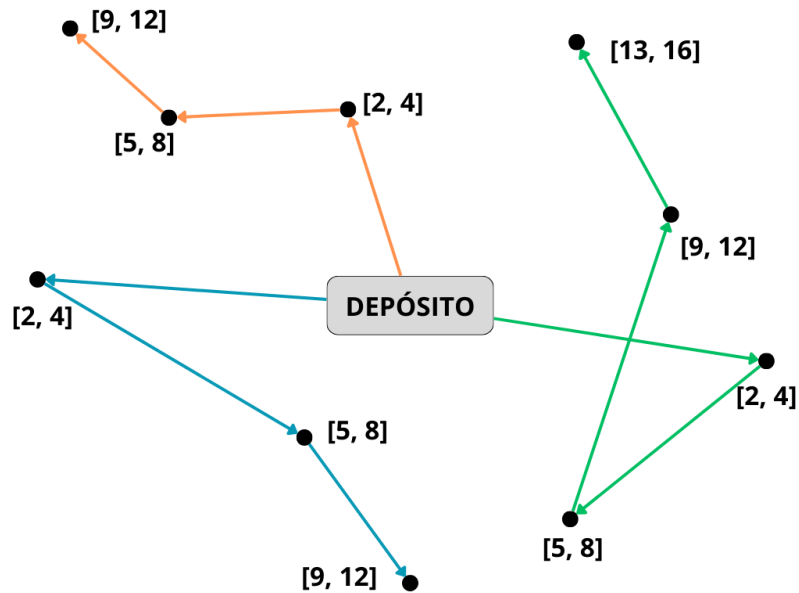


Figura 2.4: Ejemplo de un VRPTW, cada cliente tienen una ventana de tiempo  $[a, b]$  (Elaboración propia).

## 2.5. Flujo en redes

El flujo en redes es un conjunto de problemas que se encuentran en la cúspide entre varios campos, que incluyen aplicaciones matemáticas, ciencias de la computación ingeniería, administración y la investigación de operaciones [38].

Entre los principales problemas estudiados en el flujo en redes se tiene:

1. Problema de la ruta más corta (Shortest path problem). Plantea la problemática de encontrar la mejor ruta para atravesar una red desde un punto a otro de la manera más *barata* posible [38].
2. Problema de flujo máximo (Maximum flow problem). Si la red tienen capacidades en flujos de arcos, se plantea la cuestión sobre como mandar la mayor cantidad de flujo posible entre los dos puntos de una red [38].
3. Problema de flujo a costo mínimo (Minimum cost flow problem). Establece la inclusión de un costo por unidad de flujo para una red que presenta capacidades en las aristas (arcos), por lo que se desea enviar la mayor cantidad de flujo desde un punto a otro en la red al mínimo costo posible [38].

### 2.5.1. Problema de Flujo Máximo

El *problema de flujo máximo* y el *problema de la ruta más corta* son complementarios teniendo incluso cierta similitud, ya que ambos están presentes en la práctica; sin embargo estos dos problemas difieren porque modelan diferentes aspectos del *problema de flujo a costo mínimo*. El problema de la ruta más corta modela costos en las aristas pero no capacidades; por otro lado el problema de flujo máximo modela capacidades pero no costos en las aristas [38].

El problema de flujo máximo se aplica sobre una red capacitada (las aristas contienen una capacidad) en la que se desea enviar tanto flujo como sea posible entre dos vértices especiales un vértice origen  $s$  y un vértice destino  $t$  sin exceder la capacidad de cualquier arista. La formulación matemática de flujo máximo se describe como: dada una gráfica capacitada  $G = (V, A)$ , con  $V$  como el conjunto de vértices y  $A$  como el conjunto de aristas, por lo que cada arista  $(i, j) \in A$  cuenta con una capacidad no negativa  $u_{ij}$ .

Función objetivo

$$\text{máx } v \tag{2.20}$$

Sujeto a:

$$\sum_{\{j:(i,j) \in A\}} x_{ij} - \sum_{\{j:(j,i) \in A\}} x_{ji} = \begin{cases} v & \text{Para } i = s, \\ 0 & \text{Para } i \in V \setminus (\{s\} \cup \{t\}), \\ -v & \text{Para } i = t \end{cases} \tag{2.21}$$

$$0 \leq x_{ij} \leq u_{ij} \quad \forall (i, j) \in A \tag{2.22}$$

En la formulación de flujo máximo se establece un vector  $x = \{X_{ij}\}$ , que satisface tanto a la Ecuación 2.21 y 2.22, como un flujo y al valor correspondiente de la variable escalar  $v$  como el valor de dicho flujo [38]. La Ecuación 2.20 establece que se busca mandar la máxima cantidad de flujo; por otro lado la Ecuación 2.21 establece el comportamiento del flujo enviado para los vértices  $s$ ,  $t$  y los vértices intermedios, estableciendo a  $s$  como suministro y a  $t$  como demanda; la Ecuación 2.22 define la no negatividad del flujo, así como la máxima cantidad de flujo por las aristas de la gráfica.

Mediante la gráfica de la Figura 2.5 se puede observar una representación del Problema de Flujo Máximo, dada una gráfica dirigida, se observa dos nodos especiales, un nodo de origen  $s$  y un nodo destino  $t$ ; cada arco tienen un peso (flujo) [38].

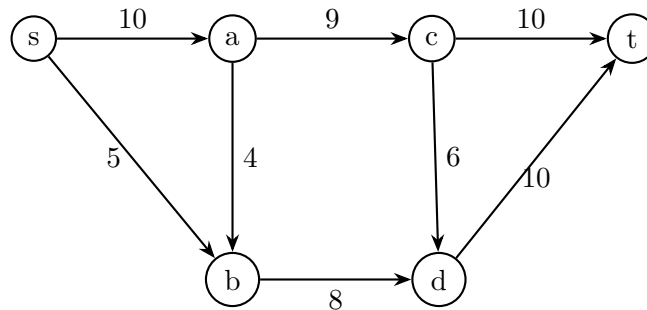


Figura 2.5: Ejemplo de red dirigida para el problema de flujo máximo (Elaboración propia).

## 2.6. Principios para una programación y diseño de rutas

La programación y el diseño de rutas para vehículos (PDRV) se define como una extensión del TSP, incorporando además restricciones reales. Por ejemplo: cada parada vehicular puede tener un volumen (de mercancías o pasajeros) que debe ser recolectado; se pueden utilizar múltiples vehículos con diferentes capacidades; y se puede establecer un tiempo máximo para el trayecto en las rutas antes de considerar un periodo de descanso, entre otras [5].

La eficiencia en la programación y el diseño de rutas por carretera puede mejorar aplicando algunos de los principios propuestos en [5], tales como:

1. Asignar a todos los vehículos paradas cuyos volúmenes estén lo más cercanos posible entre sí.
2. Ordenar los puntos de parada que requieren atención en diferentes días de manera que formen grupos más compactos.
3. Construir las rutas comenzando desde la parada más lejana respecto al depósito.
4. Disponer la secuencia de paradas en cada ruta de manera que adopte una figura tipo *lágrima*.
5. Elaborar las rutas de transporte más eficientes utilizando los vehículos con mayor capacidad disponibles.

## 2.7. Métodos de solución comunes para el Problema de Enrutamiento de Vehículos

En la siguiente sección se describen, de manera general, los métodos más utilizados para la solución a un modelo de VRP; además se describen los *solvers* más comunes para modelos de programación lineal entera. A pesar de que en este trabajo se utilizan métodos exactos, también se da una breve descripción de algunos métodos aproximados para tener un panorama más amplio sobre posibles alternativas a utilizar en el área de los modelos matemáticos relacionados al VRP.

### 2.7.1. Métodos exactos

**Programación Lineal.** La programación lineal (LP, por sus siglas en inglés) es un campo de la programación matemática cuya meta se dedica a maximizar o minimizar una *función lineal*, comúnmente llamada *función objetivo*, de tal manera que las variables presentes en dicha función están sujetas a un compendio de *restricciones* [39], por ejemplo:

Función objetivo:

$$\text{mín } f(x) = \sum_{i=1}^m \sum_{j=1}^n c_{ij}x_{ij} \quad (2.23)$$

Sujeto a:

$$\sum_{j=1}^n x_{ij} = a_i \quad (i = 1, 2, \dots, m) \quad (2.24)$$

$$\sum_{i=1}^m x_{ij} = b_j \quad (i = 1, 2, \dots, n) \quad (2.25)$$

$$x_{ij} \geq 0 \quad (2.26)$$

La Ecuación 2.23 establece una función objetivo lineal conformada por el producto de los “costos”  $c_{ij}$  y una variable de decisión  $x_{ij}$ , las Ecuaciones 2.24 y 2.25 establecen varios conjuntos de restricciones, la linealidad está establecida la Ecuación 2.26, donde se establece que  $x_{ij}$  se define que puede tomar valores reales mayores o iguales a 0.

De modo formal, la LP es una técnica matemática para la optimización de una función objetivo lineal, la cual está sujeta a un conjunto de restricciones de igualdad o desigualdad lineal [29].

**Programación Lineal Entera.** La programación lineal entera (ILP, por sus siglas en inglés) es una extensión de la LP en la que se imponen restricciones adicionales a las variables de decisión, que deben tomar valores enteros en lugar de valores continuos [29]. En otras palabras, se trata de resolver problemas de optimización donde las soluciones deben ser enteras [29].

Función objetivo:

$$\text{mín } f(x) = \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (2.27)$$

Sujeto a:

$$\sum_{i \in V, i \neq j} x_{ij} = 1, \quad \forall j \in V \quad (2.28)$$

$$\sum_{j \in V, j \neq i} x_{ij} = 1, \quad \forall i \in V \quad (2.29)$$

$$x_{ij} \in [0,1], \quad \forall i, j \in V \quad (2.30)$$

La Ecuación 2.27 establece una función objetivo lineal, conformada por un conjunto de “costos”  $c_{ij}$  haciendo un producto con una variable binaria  $x_{ij}$  (solo toma valores de 1 o 0) para cada uno de los elementos  $i$  y  $j$  pertenecientes a un conjunto  $V$ ; esta función objetivo expresa que se quiere buscar el mínimo costo (distancia, tiempo, entre otros) de entre todo el espacio de búsqueda. Las restricciones expresadas en las Ecuaciones 2.28 y 2.29 establecen que para cada  $i$  y  $j$  (considerando que  $i \neq j$ ),  $x$  tomará el valor 1. En un contexto de rutas de transporte [6], estas restricciones pueden interpretarse como la existencia de un camino de  $i$  a  $j$ . Este tipo de problemas es común en situaciones donde las variables representan cantidades discretas o indivisibles, como en la planificación de la producción, asignación de recursos, diseño de redes, entre otros [6]. La ILP se utiliza para encontrar la mejor solución posible que satisfaga todas las restricciones dadas, mientras se maximiza o minimiza una función objetivo.

**Método Simplex:** Es uno de los métodos más utilizados y populares para resolver problemas de LP. Se basa en moverse a lo largo de los vértices de la región factible para encontrar la solución óptima [29]. Es especialmente eficiente para problemas de gran tamaño y con un número moderado de restricciones [29, 38].

**Métodos de Puntos Interiores:** Estos métodos trabajan de manera diferente al método Simplex, ya que se mueven a través del interior de la región factible en lugar de los vértices, son particularmente eficaces para problemas de gran escala y son aplicables tanto a problemas de LP como a problemas no lineales [29, 38].

**Branch and Bound (Ramificación y Acotación):** Es un método fundamental para la resolución de problemas de programación entera, ya que consiste en dividir el problema original en subproblemas que son más pequeños y resolverlos, manteniendo un límite superior y un límite inferior para descartar soluciones no óptimas; este método es muy efectivo para problemas con variables enteras o binarias [38].

**Branch and Cut (Ramificación y Corte):** Es una extensión del método Branch and Bound que añade cortes (restricciones adicionales) a los subproblemas para eliminar regiones no factibles sin explorar. Esto mejora la eficiencia del proceso de búsqueda [38].

## Solvers de ILP

La mayoría de los lenguajes de programación modernos utilizados en ciencias de datos como Julia y Python se comunican con solvers ya sea de LP, ILP o MILP (Mixed Integer Linear Programming) mediante bibliotecas que realizan la función de interfaces, ya que dichos solvers se encuentran desarrollados en lenguajes de programación compilados y con una mayor gestión de memoria y recursos como lo es el lenguaje C++. Para el lenguaje python una de las bibliotecas para dicha comunicación es PuLP, mientras que para el lenguaje Julia se cuenta con la biblioteca JuMP [40].

Entre los principales solver para resolver modelos de ILP se encuentran los siguientes:

- **CBC (Coin-or branch and cut):** Es un solver de código abierto que utiliza Branch and Cut [41].
- **GLPK (GNU Linear Programming Kit):** Este solver ofrece el uso del método Simplex y del método de puntos interiores [40, 42].
- **Gurobi y CPLEX:** Son solvers comerciales que implementan algoritmos avanzados para la LP y ILP [40, 43, 44]. Pueden proporcionar soluciones mucho más rápidas en comparación con solvers de libre uso como GLPK y CBC [40].

La elección del método y el solver adecuado dependerá de la naturaleza del problema específico, su tamaño, y las restricciones involucradas, para este trabajo se utilizó el solver CBC, por lo que se describen sus características.

### Solver COIN-OR Branch and Cut

CBC es un solver de LP y ILP de código abierto que implementa métodos de *ramificación y acotación* (Branch and Bound) y *ramificación y corte* (Branch and Cut) [41]. Entre sus características se puede mencionar:

- **Configuración de Parámetros:** Permite establecer diferentes parámetros de solución, como límites de tiempo, tolerancias y opciones de salida, lo que te permite personalizar el comportamiento del solver.
- **Interfaz de Ejecución:** Proporciona métodos para ejecutar el solver y obtener resultados.
- **Salida y Reportes:** Controla el nivel de detalle de la salida generada por el solver, lo que permite ajustar la información que se imprime durante la solución.

### 2.7.2. Métodos aproximados

Las heurísticas y metaheurísticas son técnicas utilizadas para aproximar una solución de problemas complejos de manera eficiente, especialmente cuando no es factible encontrar una solución óptima en un tiempo computacionalmente razonable. Estas técnicas son

comúnmente utilizadas en el campos como: la optimización, la inteligencia artificial, la planificación y la toma de decisiones [32].

**Heurística.** Es un método de exploración que explota algunos aspectos específicos del problema en cuestión y que solamente aplica a éste [32]. Las heurísticas son útiles cuando se necesita encontrar una solución aceptable en un tiempo limitado, aunque no necesariamente la mejor solución. Algunos ejemplos de heurísticas comunes incluyen el “algoritmo voraz”, las reglas de búsqueda en árboles de decisión [45]. Para la construcción de rutas de transporte, se hace mención dos métodos heurísticos muy conocidos:

- Método “de ahorros”: Conocido también como de valoración de ahorros Clarke-Wright, es un método flexible cuyo objetivo es minimizar la distancia total recorrida por todos los vehículos, a la vez que se minimiza indirectamente el número de vehículos necesarios para atender todos los puntos de parada [5].
- Método “de barrido”: Es un método sencillo de aplicar, incluso puede realizarse mediante cálculos manuales, consiste solamente de dos etapas: en la primer etapa las paradas se asignan a los vehículos disponibles y en la segunda etapa se establece la secuencia de las paradas dentro de cada una de las rutas. A pesar de ser sencillo, tienen un problema debido a las dos etapas que tiene, ya que la sincronización y el tiempo total empleado en algunas rutas no se encuentran bien determinados y manejados [5].

**Metaheurística.** Es un método de exploración general, algunas veces estocástico (sometido al azar), que aplica de la misma forma a diferentes problemas, algunos ejemplos son la búsqueda Tabú, recocido simulado, colonia de hormigas y algoritmos evolutivos [32]. Básicamente una metaheurística se define también como un método heurístico, pero a diferencia de este último, se utiliza para resolver algún problema computacional con un enfoque más general; utiliza parámetros que son dados por el usuario para aplicarse en procedimientos genéricos y abstractos de una manera eficiente. Comúnmente se suele implementar sobre problemas que no tienen un método heurístico específico y por consiguiente no se puede hallar una solución aceptable; o bien cuando no es factible aplicar algún método exacto. Algunos ejemplos de metaheurísticas populares son:

- Algoritmo de búsqueda tabú (TS): Evita movimientos repetitivos en la búsqueda y mantiene un registro de soluciones visitadas anteriormente [45, 46].
- Algoritmo genético (GA): Utiliza conceptos inspirados en la evolución biológica, como selección, mutación y cruza, para buscar soluciones en un espacio de búsqueda [45].
- Recocido simulado (SA): Simula el proceso de enfriamiento de un material para explorar el espacio de búsqueda de manera eficiente [45].

- Algoritmo de enjambre de partículas (PSO): Modela el comportamiento de un enjambre de partículas en busca de la mejor solución [32].
- Optimización por colonia de hormigas (ACO): Está inspirado y enfocado en el comportamiento de las hormigas en la búsqueda de caminos (rutas) óptimos [47].



---

# Capítulo 3

## Estado del arte

---

La literatura de VRP tiende a ser muy basta, las categorías aplicables al caso de estudio como son: VRPTW y SBRP (School Bus Routing Problem), son un pilar para el trabajo de investigación planteado, los artículos más relevantes del tema a trabajar se encuentra con Moradi [48] quien establece un nuevo algoritmo para un modelo matemático multi-objetivo para direccionar un VRPTW, establecen dos funciones objetivo enfocadas en minimizar, una en el aspecto de reducir la cantidad de vehículos ( $k$ ) y otra en la selección del conjunto de rutas más cortas; el conjunto de restricciones propuestas son las fundamentales para el modelado clásico de VRPTW. La solución al modelo propuesto, es mediante el uso de un MODLEM (Multi-Objective Discreet Learnable Evolution Model), una mejora de LEM (Learnable Evolution Model) el cual es un método de machine learning, que permite direccionar el VRPTW multi-objetivo, sus resultados experimentales y comparaciones indican que MODLEM tienen mejoras en la eficiencia al ser un enfoque de enrutamiento inteligente.

Li et al. [3] establecen un modelo de VRP aplicado a un ámbito del tráfico urbano, dónde se describen un conjunto de situaciones relacionadas a la contaminación del medio ambiente, debido al incremento del número de viajes en automóvil. Dadas todas estas circunstancias proponen un modelo de optimización multi-objetivo que maximice la utilidad del transporte. Para poder establecer el modelo matemático se consideraron diversos aspectos de costos para cada tipo de medio de transporte (bicicleta, metro, autobús, taxi y automóvil), para cada uno de estos le dispusieron un costo y métricas de estudios previos. Su propuesta utiliza fuertemente el área de IO, esto es notorio para la determinación de costos de cada uno de los medios de transporte propuestos, en cuanto a la solución se utilizan y comparan los métodos de punto ideal, secuencia jerárquica y ponderación lineal; en donde determinaron que los mejores resultados fueron dados por el método del punto ideal. La mayor relevancia del problema es en todo el análisis con IO para establecer las funciones multi-objetivo del problema, los métodos usados para solucionar el modelo fueron comparados reportando el

rendimiento de las pruebas con los métodos mencionados.

Zhang et al. [47] proponen una combinación de dos heurísticas denominada IFACO (improved hybrid fish swarm-ant colony algorithm), que combina al algoritmo de enjambre de peces artificiales (AFSA) y el algoritmo de optimización por colonia de hormigas (ACO); todo con el fin de solucionar la categoría VRPTW. Probaron su modelo mediante el conjunto de datos estándar de “Solomon” el cual tiene la característica de tener los puntos de los clientes agrupados, ventanas de tiempo ajustadas y capacidades de vehículos pequeñas. Realizaron una comparativa del algoritmo IFACO, I-PSO y S-PSO-VRPTW, usando diferentes tamaños de la colección de datos de Solomon; para ello estipularon ciertos parámetros: el valor óptimo de la distribución de la distancia (TD), valor óptimo del número de vehículos (NV), porcentaje mejorado en la calidad de la solución (GAP).

Con las pruebas realizadas, los autores determinaron que la propuesta de IFACO tienen a converger rápidamente cuando el número de iteraciones esta ente 0 y 70, pero el algoritmo converge lentamente después de 70 iteraciones. A pesar de esto el algoritmo tiende a converger hacia la respectiva solución óptima. En general IFACO tiene una buena convergencia y una capacidad óptima de búsqueda y puede evitar el “óptimo local”. El algoritmo de enjambre de peces artificiales da un grado de “hacinamiento”, lo que permite mejorar la transición de estado para actualizar la “feromona global”, además se diseñaron tres operadores: conmutación de enrutamiento de 2-opt, conmutación de enrutamiento y enrutamiento cruzado, como estrategias de búsqueda de vecindario para ampliar el rango de búsqueda y mejorar la calidad de la solución.

S.W. et al. [45] plantean como objetivo probar un nuevo algoritmo heurístico llamado *Harris Hawks Optimiser* (HHO) en un caso de estudio para generar rutas entre los distritos de Kuantan y Pekan ubicados en Pehang, Malasia. El proceso que utilizaron consistió en plantear un modelo matemático, considerando una función objetivo simple enfocada en la búsqueda de la distancia mínima en un conjunto de 52 puntos con sus respectivas ventanas de tiempo y generar un número de rutas respetando ciertas restricciones. Las restricciones que plantearon tuvieron un enfoque en las ventanas de tiempo para cada cliente, tiempo de servicio y una demanda; además para cada vehículo se les asignó restricciones relacionadas a su capacidad.

La matriz de datos (distancias) que utilizaron fue convertida a una matriz de tiempos y ordenada aleatoriamente, además establecieron una ventana de tiempo de 120 minutos para que un vehículo termine una ruta completa, es decir, que salga del depósito y retorne a este mismo. Se realizó una comparación contra otras tres heurísticas ya aplicadas a este tipo de problemas: Artificial Bee Colony (ABC), Firefly Algorithm (FA), Particle Swarm Optimisation (PSO) [45]. Además de estos, también se hicieron pruebas con otros algoritmos relativamente nuevos: Moth Flame Optimiser (MFO), Grey Wolf Optimiser (GWO), y Whale Optimisation Algorithm (WOA) [45]. Para realizar estas comparativas, se tomaron como parámetros: Average Fitness, Desviación estándar y Best Fitness.

Con el fin de evaluar el significado de los resultados utilizaron una prueba estadística no parametrizada llamada *Wilcoxon-signed rank test*, la cual fue usada con HHO y los

otros algoritmos a comparar. Los resultados indican que los algoritmos HHO y WOA son adecuados para buscar el *fitness mínimo*, que es el número de vehículos a ser asignados. Además el HHO obtuvo el mejor *fitness medio* en comparación a los otros seis algoritmos probados. Los resultados obtenidos de la simulación determinan que la solución óptima usando HHO tiende a ser consistente al tener la mejor oportunidad de no romper la ventana de tiempo y el máximo tiempo de viaje. A pesar del mejor desempeño del algoritmo HHO en el contexto de VRPTW propuesto, en el modelo matemático que describen respecto al caso analizado, no se utilizaron las restricciones de capacidad para el proceso de solución del modelo con HHO, esto es notorio ya que en los datos utilizados para alimentar el modelo, no se contempla el dato de la *demanda* para cada uno de los 52 puntos, solo se utilizaron ventanas tiempo y tiempo de servicio; además de señalar que el caso de estudio es una instancia con pocos puntos.

Liu y Wang [1] proponen un algoritmo denominado Modified Adaptive Large Neighborhood Search with Nearest Vehicle Dispatched (MALNSN) siendo éste una extensión del Adaptive Large Neighborhood Search algorithm (ALNS), todo esto bajo un contexto de un transporte en caso de emergencias cuando el transporte ferroviario presente fallas y los usuarios queden varados. Se utilizaron datos simulados, tomados del Solomon's Benchmark, los datos fueron tomados para comparar principalmente el algoritmo MALNSN en contra del ALNS (algoritmo original sin modificaciones). Propusieron una función multi-objetivo que se compone de dos partes (Bus y Passenger), donde se manejan parámetros  $\alpha$  y  $\beta$ , que permiten dar una ponderación.

Para cada algoritmo realizaron la simulación 10 veces y para cada una de esas repeticiones, se calculó el valor de la función objetivo. Por cada 10 experimentos, cada 50 iteraciones se registraron un total de 500. El algoritmo MALNSN redujo el valor objetivo en 16.96 %, 11.59 %, 8.98 % 7.01 % y 12.80 % comparado con los algoritmos SA, TS, PSO-GA (Particle Swarm Optimization with Genetic algorithm), PSO-ACO (Particle Swarm Optimization with Ant Colony Optimization) y ALNS respectivamente. Señalan que de acuerdo a los resultados experimentales al inicializar los parámetros de la función objetivo con  $\alpha = 0.6$  y  $\beta = 0.8$ , se puede obtener una mejor solución.

Su caso de estudio, enfocado a un transporte de emergencia, uso un proceso de agrupamiento en tiempo real, con el uso de un algoritmo de clustering mejorado y adaptativo que parte de DBSCAN. El nuevo algoritmo propuesto denominado MALNSN tiene un mejor desempeño principalmente en comparación del ALNS (versión original) así como también una mejora en comparación con otros algoritmos. La propuesta establecida en el trabajo es adecuada al caso de estudio planteado, pero el modelo matemático tiene algunas deficiencias en la definición de los conjuntos establecidos, errores en la escritura de los parámetros y variables que el mismo modelo utiliza (errores de nomenclatura), lo que imposibilita la recreación experimental de sus procedimientos; lo favorable de la investigación radica en que utilizan algoritmos de clustering para la determinación de paradas vehiculares para el traslado de pasajeros, así como la determinación de una función multi-objetivo tomando en consideración costos en el ámbito del pasajero. La propuesta de método mejorado MALNSN

obtuvo una ganancia promedio de 17.11 % mayor a la del algoritmo original ALNS, por lo que las mejoras propuestas son eficientes.

Sciortino et al. [37] describen las características de un School Bus Routing Problem (SBRP), que es un problema del mundo real compuesto por varios subproblemas (selección de paradas de autobús, generación de rutas de autobús, entre otros), añaden varias características realistas, como la elegibilidad de los estudiantes, distancias máximas para caminar, selección de paradas de autobús, tiempos máximos de viaje, diferentes tipos de autobuses, paradas múltiples y tiempos de permanencia de los autobuses. Proponen un algoritmo heurístico basado en un enfoque de búsqueda local iterada. Los objetivos principales son minimizar la cantidad de autobuses utilizados, la distancia total que caminan los estudiantes y el tiempo total de viaje de la ruta. Sus resultados indican que el algoritmo encuentra soluciones de alta calidad en muy poco tiempo de cálculo. En este trabajo se tienen estudios estadísticos previos para determinar la distancia adecuada para establecer puntos de parada considerando las ubicaciones de los alumnos, lo que demuestra un trabajo previo antes de plasmar el modelo matemático del problema, concluyen que su propuesta puede ser probada con hasta 1,800 paradas de autobuses potenciales y 750 alumnos por lo que se puede considerar una instancia lo suficientemente grande para un caso de estudio.

Ha-Bang Ban et al. [49] describen un modelo basado en el término nombrado School Bus-Driven Problem with Resource Constraints (SBDP-RC) como un problema de optimización. Plantean como objetivo encontrar un esquema que minimice la suma de todos los tiempos de espera de los alumnos, proponen como método de solución un método denominado “Hybrid Variable Neighborhood Search” que es una metaheurística de dos fases para resolver el problema, en la primera fase usan una heurística de inserción y en la siguiente fase implementan una mejora de la solución con una “Variable general de búsqueda de vecindad” (GVNS), con una búsqueda aleatoria de vecindarios combinada con la técnica de agitación. A pesar de que la propuesta toma el contexto del transporte de personas (alumnos), se enfoca en optimizar los tiempos de espera de los alumnos en las paradas de autobús, por lo que no toma en consideración otro tipo de aspectos plasmadas como restricciones en el contexto del VRP, su propuesta de solución es factible para 80 clientes (ubicaciones de los alumnos), pero para un caso práctico no es conveniente, por lo que el método propuesto no es apropiado para instancias grandes.

Los trabajos a modo de revisión sobre clasificaciones, modelos, algoritmos relacionados al VRP se encuentran primeramente con los realizados por Zhang et al. [50], quienes profundizan con mayor detalle sobre distintos tipos de modelos de VRP, junto a sus clasificaciones. Por otro lado, Montoya-Torres et al. [51] abordan el tema del VRP con depósitos múltiples (MDVRP). El-Sherbeny et al. [52] dan una revisión sobre los métodos utilizados para dar solución a un VRPTW. En la Tabla 3.1, se resumen los trabajos más relevantes consultados, así como sus propuestas de solución. Estos fueron elegidos debido a su parecido tanto visual como del modelo matemático a tratar en este trabajo basado en un caso de estudio enfocado a transporte urbano.

Tabla 3.1: Resumen de la información de los artículos consultados (Elaboración propia).

<b>Año</b>	<b>Autor</b>	<b>Tipo de modelo</b>	<b>Tipo de función objetivo</b>	<b>Método de solución</b>
2020	Moradi [48]	MO-VRPTW	Multi-objetivo	MODLEM
2021	Li et al. [3]	VRP	Multi-objetivo	The ideal point method
2022	Zhang et al. [47]	VRPTW	Simple	IFACO
2022	S.W. et al. [45]	VRPTW	Simple	HHO
2022	Liu y Wang [1]	VRPTW	Multi-objetivo	MALNSN
2022	Sciortino et al. [37]	SBRP	Simple	Heurística basada en un enfoque de búsqueda local iterada
2023	Ha-Bang Ban et al. [49]	SBDP-RC	Simple	Hybrid Variable Neighborhood Search
2023	Department of Industrial Engineering, Yildiz Technical University, İstanbul, Turkiy [4]	PSRP	Simple	ILP

El tema de clustering aplicado explícitamente al área de transporte es implementado por Villalba et al. [53] mediante una comparativa entre los algoritmos DBSCAN y k-means para clasificar clientes mediante coordenadas geográficas para un caso de VRP, la base de datos de clientes que utilizaron fue considerablemente amplia de un total de 2.2 millones de registros, su objetivo fue exportar la mayor cantidad de clientes representativos; muestran la importancia de reducir la cantidad de datos para un VRP, sin embargo no hacen un trazado de rutas con algún modelo, solo se realizó la comparativa de dos algoritmos de clustering mediante el número de clusters formados, la función de distancia y el porcentaje de reducción de datos; la comparativa carece de algún criterio mediante métricas adecuadas para determinar que algoritmo de clustering es mejor bajo métricas de calidad.

Liu y Wang [1] utilizaron un proceso de agrupamiento en tiempo real, con el uso de un algoritmo de clustering DBSCAN mejorado y adaptativo, los resultados determinados del proceso del algoritmo mejorado DBSCAN lo utilizaron para alimentar un modelo de ILP para un caso de estudio enfocado en el transporte ferroviario China, no requirieron valorar la calidad de los clusters formados ya que el parámetro *Eps* de DBSCAN se auto-calcula.

La comparativa realizada por M. Sánchez [7] mostró las ventajas de utilizar algoritmos basados en densidad como: DBSCAN, HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) y OPTICS (Ordering Points to Identify the Clustering Structure), así como de K-means, para determinar paradas vehiculares usando coordenadas geográficas en formato de latitud y longitud a una distancia de 1 km, mencionando la importancia de la configuración adecuada de los parámetros de los algoritmos basados en densidad, además de aplicar métricas de calidad de clusters para determinar que algoritmo arroja mejores resultados, sin embargo solo utiliza dos características para sus procesos de clustering al utilizar una métrica de distancia Haversine.

Entre las implementaciones de la envolvente convexa al VRP se encuentra S. S. Néia et al. [54] quienes propusieron una sugerencia para valorar la calidad de las rutas obtenidas en un VRP, mediante un análisis geométrico con una envolvente convexa para la determinación del tamaño de un área propuesta.

B. Kallehauge [55] dió una revisión de algoritmos exactos, retomando las propuestas de solución partiendo desde el TSP y mTSP hasta el VRPTW. Primeramente retoma el enfoque de Dantzing para la identificación de desigualdades lineales o planos de corte que describen el politopo definido por la envolvente convexa en un espacio euclidiano que representan el conjunto de las soluciones factibles para el problema de optimización combinatoria; se hace mención de la relajación y dualidad Lagrangiana que utiliza una envolvente convexa de los vectores de incidencia de soluciones.

T.K. Ralphs et al. [56] describieron los aspectos más relevantes del algoritmo *branch and cut*, ya que diseñan subrutinas para separar efectivamente un punto fraccionario dada la envolvente convexa. Resaltan que el aspecto más importante y desafiante es el diseñar subrutinas que separen efectivamente un punto fraccionario dado de la envolvente convexa de soluciones enteras al aplicarse el *branch and cut* al VRP. Su algoritmo de *branch and cut* tienen un buen rendimiento mejorando los tiempos de solución.

J. M. Belenguer et al. [57] consideraron una versión de un VRP denominado *Problema de Rutas de Vehículos de Entrega Dividida* (SDVRP), donde definieron una solución factible al problema mediante una envolvente convexa de los vectores de incidencia asociados. Propusieron un algoritmo de plano de corte, que permitió la resolución de instancias pequeñas para el problema SDVRP, pero para instancias grandes los valores enteros se obtuvieron mediante el algoritmo branch and bound; además aseveran que aun no es viable el método porque la formulación utilizada es una relajación del problema original.

Hao Tang et al. [58] propusieron un método heurístico interactivo para resolver un VRP, donde expresan la importancia del aspecto visual de las rutas en aplicaciones prácticas de transporte de carga, donde la envolvente convexa la utilizaron como una métrica para valorar la calidad de las soluciones determinadas por su método heurístico.

Mir Mohammad et al. [59] desarrollaron un método heurístico de búsqueda local, basado en la inserción más cercana de construcción de la envolvente convexa para un TSP, denominado método de inserción más cercano en la búsqueda local de envolvente convexa (NICH-LS). Su heurística se probó en numerosas instancias TSP tomadas del TSPLIB, los cuales produjeron resultados aceptables, cercanas a la solución óptima en tiempos computacionales aceptables.

M.A. Sánchez y L. Loza [8] utilizaron una envolvente convexa para identificar los puntos más alejados para establecerlos como *terminales* para un CVRP aplicándolo a cuatro escenarios, obteniendo siempre los puntos más alejados a cada uno de los depósitos establecidos en cada escenario. Su propuesta no considera flujo para el traslado de alumnos hacia destinos particulares, sin embargo la propuesta permite obtener rutas ópticas con un solver libre COIN-OR Branch and Cut adecuado para el modelo de ILP que propusieron. En sus escenarios la envolvente convexa demostró ser una propuesta adecuada para obtener los puntos terminales, en contra posición a otros métodos heurísticos como lo es el algoritmo Dijkstra.



---

# Capítulo 4

## Marco metodológico

---

Retomando la metodología propuesta de la sección 1.5, se muestra nuevamente el gráfico de bloques para definir el marco metodológico.

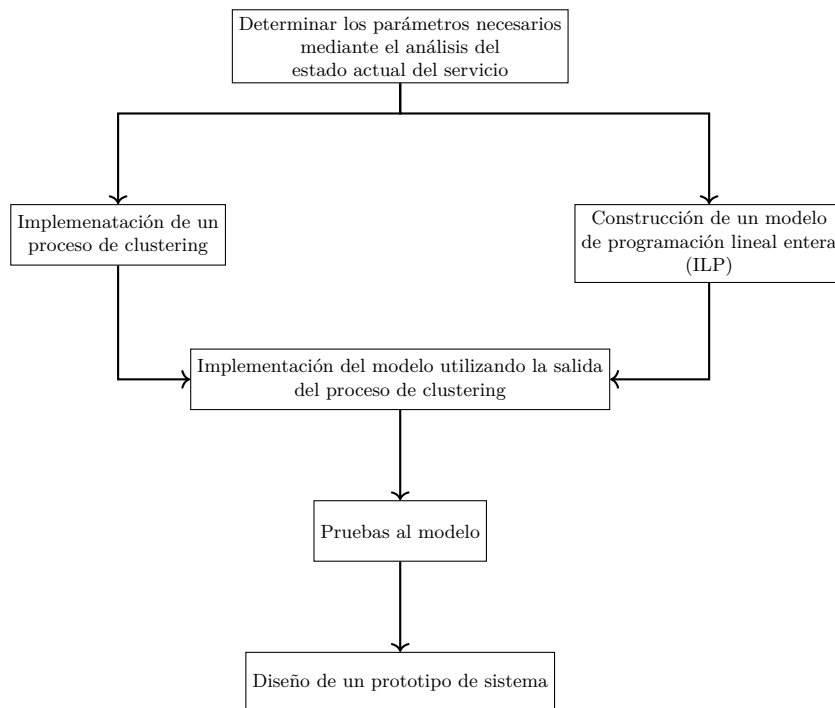


Figura 4.1: Diagrama de bloques de la metodología (Elaboración propia).

En este capítulo se describen los componentes a utilizar para esta investigación, los métodos propuestos se retoman de [1], donde utilizaron procedimientos por etapas, partiendo desde un proceso de clustering, formulación del modelo matemático, solución del modelo y una comparativa. Se añadieron etapas de estudios cuantitativos y análisis de datos al igual que Sciortino et al. [37]. A continuación se describen en el orden a ser presentados.

## 4.1. Estudio cuantitativo

Debido a la naturaleza del problema, se utilizó un enfoque cuantitativo para desarrollar el modelo matemático considerando el caso en estudio, planteado en la sección 1.1, siguiendo las características que suelen tomarse del VRP; el estudio cuantitativo consistió en el análisis de los datos de la afluencia del servicio proporcionado por la dirección de transporte, en conjunto de una encuesta y el análisis de la matrícula escolar de alumnos, para estimar la cantidad de usuarios del servicio y su distribución. Se describió el contexto actual del enrutamiento del servicio, incluyendo datos sobre las rutas existentes, la cantidad de estudiantes, las ubicaciones de las paradas y los espacios académicos a considerar.

## 4.2. Procedimientos de análisis de datos

Los datos trabajados se conformaron por la matrícula de alumnos, los datos de la encuesta y la ubicación de los espacios académicos de la UAEMex; estos datos fueron filtrados considerando los siguientes aspectos:

1. Eliminación de registros duplicados o incompletos, en los campos de código postal (CP) y dirección, en la matrícula de alumnos.
2. Identificación el rango de códigos postales para establecer la zona de servicio de transporte, donde se requieren identificar posibles paradas de autobús.
3. Realización de un filtrado aplicado a los datos de la matrícula de alumnos y obtener aquellos registros con el rango de CP.
4. Generación de un archivo con la conversión de CP a coordenada geográfica en formato (longitud, latitud), mediante la consulta de APIs (Application Programming Interface) de servicios de rutas y mapas como la de OpenStreetMap [60], Códigos Postales de México y Servicios de Google.

## 4.3. Aplicación de procesos de clustering

Al igual que [1] se utilizó el algoritmo de clustering DBSCAN, descrito en la sección 2.1, sin modificaciones propuestas por algunos autores al algoritmo original, ya que solo se

requiere un proceso de clustering sin que el algoritmo estime automáticamente la distancia del proceso de clusterizado; estableciendo los parámetros y métricas de distancia en un rango recomendado por [16, 37]. Además se aplican métricas de valoración de calidad para analizar los clusters formados.

#### **4.4. Modelo matemático**

Se desarrollaron modelos de optimización de ILP, basados en VRP y sus variantes para poder modelar el caso de estudio, siguiendo los modelos teóricos básicos tanto de VRP, así como de flujo máximo descritos en el Capítulo 2 en la sección 2.7.

#### **4.5. Solución del modelo con métodos exactos y aproximados**

Con lo descrito en las secciones 2.7.1 y 2.7.2, el modelo de ILP pudo ser resuelto con métodos exactos. Se utilizó un solver de ILP descrito en la sección 2.7 para resolver el modelo de ILP del caso de estudio y tener un panorama de la complejidad del mismo.

#### **4.6. Implementación de un prototipo de sistema**

Todos los componentes trabajados: análisis de datos, clustering, modelo matemático y la solución al modelo mediante un solver, se implementaron en un sistema web con la capacidad de trazar las rutas correspondientes sobre un mapa y determinar la viabilidad de las nuevas rutas formadas.



---

# Capítulo 5

## Análisis de resultados

---

Mediante la metodología propuesta, ver sección 1.5, se establecieron diversas etapas para un proceso experimental, con la intención de obtener un conjunto de rutas para el servicio de transporte del potrobús, mediante el análisis de la matrícula de alumnos de la UAEMex. Las etapas del proceso experimental fueron la siguientes:

1. Preprocesamiento de datos.
2. Procesos de clustering con el algoritmo DBSCAN.
3. Modelos matemático del caso de estudio.
  - Implementación del modelo mediante un solver de ILP
4. Visualización de las rutas.

Dentro de las siguientes secciones de este capítulo se describe cada una de las etapas del proceso experimental utilizado, detallando principalmente el tratamiento, volumen y comportamiento de los datos.

### 5.1. Preprocesamiento de los datos

Aplicar un preprocesamiento de datos es necesario para cualquier conjunto de datos, sin importar si es de alguna instancia o si es del caso de estudio planteado. Con el preprocesamiento de datos se pueden determinar ciertos parámetros esenciales para un modelo, así como descartar datos innecesarios que puedan alterar el resultado del proceso de clustering o del resultado del modelo matemático.

### 5.1.1. Descripción de los conjuntos de datos y su función

En esta etapa se utilizaron diversos conjuntos de datos tanto para el análisis, así como para determinar los parámetros del modelo matemático, concerniente al caso de estudio, los datos estaban distribuidos en cuatro archivos diferentes, los cuales se conformaron por:

- Matrícula de alumnos en formato CSV (comma-separated values) que contienen los campos (columnas): *ALUMNO*, *EA* (*Espacios Académicos*), *CALLE*, *POBLACION*, *ESTADO*, *MUNICIPIO* y *CP*.
- Un archivo con los EA, es decir, Facultades y Preparatorias en donde los alumnos realizan sus estudios en formato JSON (JavaScript Object Notation), cada EA tiene su correspondiente coordenada geográfica y su CP.
- Un archivo con los códigos postales (CP) por municipio en formato JSON.
- Un archivo con la correspondencia de CP a coordenada geográfica (longitud y latitud) en formato CSV, que tienen la función de ser una base de datos de códigos postales.

Además de los archivos mencionados, también se implementó un quinto archivo para lograr la simulación de los horarios en que los alumnos entran y salen de sus respectivos EA, con la consideración de si los alumnos estudian en nivel de Bachillerato o Licenciatura, con el fin de realizar consultas por horarios específicos. Se propuso un grupo de horarios definidos en un pequeño conjunto de datos en un archivo en formato JSON, cuyos valores son mostrados en la Tabla 5.1.

Tabla 5.1: Horarios de entrada propuestos (Elaboración propia).

Tipo de horario	Valores
Entrada Licenciatura	[7:00,9:00,11:00,13:00,15:00,17:00,19:00]
Salida Licenciatura	[9:00,11:00,13:00,15:00,17:00,19:00, 21:00]
Entrada Bachillerato	[7:00, 2:00]
Salida Bachillerato	[13:00, 20:00]

### 5.1.2. Preprocesamiento de la matrícula de alumnos

Los datos de la matrícula de alumnos se sometieron a un proceso de “limpieza”, eliminando los registros con datos incompletos o faltantes; de este modo la cantidad de registros pasó de 96,604 a 96,113 registros, teniendo una pérdida o depuración de 491 registros, equivalente a un 0.5 % de los datos. En la Tabla 5.2 se puede ver la distribución de los alumnos hacia los 52 EA principales de la UAEMex. Al analizar la distribución de los alumnos por EA, se observó que el EA con mayor volumen de alumnos es la *Facultad de Contaduría y Administración*, seguido por la *Facultad de Medicina*.

Tabla 5.2: Distribución de la totalidad de alumnos por EA (Elaboración propia).

ID	EA	Cantidad
1	FACULTAD DE CONTADURIA Y ADMINISTRACION	6313
2	FACULTAD DE MEDICINA	5191
3	CENTRO UNIVERSITARIO UAEM TEXCOCO	5051
4	CENTRO UNIVERSITARIO UAEM VALLE DE MEXICO	3843
5	PLANTEL LIC. ADOLFO LOPEZ MATEOS DE LA ESCUELA PREPARATORIA	3592
6	PLANTEL NEZAHUALCOYOTL DE LA ESCUELA PREPARATORIA	3542
7	PLANTEL CUAUHEMOC DE LA ESCUELA PREPARATORIA	3269
8	FACULTAD DE DERECHO	3176
9	CENTRO UNIVERSITARIO UAEM VALLE DE CHALCO	3164
10	UNIDAD ACADÉMICA PROFESIONAL CHIMALHUACAN	3016
11	FACULTAD DE CIENCIAS DE LA CONDUCTA	2953
12	PLANTEL DR. ANGEL MA. GARIBAY KINTANA DE LA ESCUELA PREPARATORIA	2805
13	FACULTAD DE INGENIERIA	2740
14	CENTRO UNIVERSITARIO UAEM ZUMPANGO	2553
15	PLANTEL TEXCOCO DE LA ESCUELA PREPARATORIA	2389
16	PLANTEL SOR JUANA INES DE LA CRUZ DE LA ESCUELA PREPARATORIA	2177
17	PLANTEL DR. PABLO GONZALEZ CASANOVA DE LA ESCUELA PREPARATORIA	2170
18	FACULTAD DE ARQUITECTURA Y DISEÑO	2084
19	PLANTEL IGNACIO RAMIREZ CALZADA DE LA ESCUELA PREPARATORIA	1984
20	CENTRO UNIVERSITARIO UAEM AMECAMECA	1972
21	UNIDAD ACADÉMICA PROFESIONAL CUAUTITLAN IZCALLI	1941
22	FACULTAD DE ECONOMIA	1930
23	CENTRO UNIVERSITARIO UAEM ECATEPEC	1862
24	CENTRO UNIVERSITARIO UAEM NEZAHUALCOYOTL	1804
25	FACULTAD DE ENFERMERIA Y OBSTETRICIA	1768
26	UNIDAD ACADÉMICA PROFESIONAL TIANGUISTENCO	1632
27	CENTRO UNIVERSITARIO UAEM ATLACOMULCO	1553
28	CENTRO UNIVERSITARIO UAEM VALLE DE TEOTIHUACAN	1375
29	FACULTAD DE CIENCIAS POLITICAS Y SOCIALES	1371
30	FACULTAD DE QUIMICA	1351
31	FACULTAD DE CIENCIAS	1345
32	FACULTAD DE HUMANIDADES	1235
33	FACULTAD DE TURISMO Y GASTRONOMIA	1225
34	PLANTEL ISIDRO FABELA ALFARO DE LA ESCUELA PREPARATORIA	1183
35	FACULTAD DE CIENCIAS AGRICOLAS	1150
36	FACULTAD DE LENGUAS	1139
37	CENTRO UNIVERSITARIO UAEM TEMASCALTEPEC	1017
38	CENTRO UNIVERSITARIO UAEM TENANCINGO	974
39	UNIDAD ACADÉMICA PROFESIONAL ACOLMAN	870
40	FACULTAD DE MEDICINA VETERINARIA Y ZOOTECNIA	861
41	FACULTAD DE ODONTOLOGIA	806
42	FACULTAD DE ARTES	670
43	UNIDAD ACADÉMICA PROFESIONAL HUEHUETOCA	628
44	FACULTAD DE GEOGRAFIA	617
45	FACULTAD DE PLANEACION URBANA Y REGIONAL	592
46	UNIDAD ACADÉMICA PROFESIONAL TEJUPILCO	435
47	ESCUELA DE ARTES ESCENICAS	379
48	UNIDAD ACADÉMICA PROFESIONAL TLALNEPANTLA	324
49	FACULTAD DE ANTROPOLOGIA	247
50	PLANTEL MTRO. JOSE IGNACIO PICHARDO PAGAZA DE LA ESCUELA PREPARATORIA	246
51	INSTITUTO INTERAMERICANO DE TECNOLOGIA Y CIENCIAS DEL AGUA	46
52	INSTITUTO DE CIENCIAS AGROPECUARIAS Y RURALES	39

La información de la matrícula de alumnos, concerniente a la ubicación, tuvo que ser completada para determinar la zona donde habitaban los alumnos. A pesar de que se disponía de datos domiciliarios como: colonia y calle, debido a las variaciones presentes en estos campos, no fue viable utilizarlo para determinar dicha ubicación por lo que se utilizó el campo de CP para determinar este elemento. Al utilizar CP, se aplicó un primer filtro para eliminar aquellos CP que no estaban formados para el Estado de México, es decir, aquellos que iniciaban con un valor distinto a 5 y su longitud no es de 5 dígitos; después se realizó un tercer filtrado para obtener aquellos registros que estaban en la *zona del servicio* de transporte.

Para determinar la *zona de servicio*, se dispuso de un conjunto de datos complementarios, los cuales se conformaron por los CP de municipio aledaños, por lo que se definió a la *zona de servicio* como el compendio de los 10 municipios cercanos al campus principal donde el servicio de transporte opera en la actualidad. Esta zona se conformó por los municipios de: Toluca de Lerdo, Zinacantepec, Metepec, Lerma, Ixtlahuaca, Tenango, Santiago Tianguistenco, Temoaya, Ocoyoacac y Xonacatlán. Cada uno de estos municipios contó con un número de CP. La cantidad CPs por municipio es variada, debido al tamaño que ocupa cada municipio y su división de área establecida por parte de Correos de México [61]. El rango de CP, para la zona de servicio, se conformó desde el 50000 al 52757; con este filtrado más específico, los registros pasaron de 96,113 a 39,700, por lo que no se consideraron 56,413 registros, equivalente a un 41.3% de los datos.

Los 39,700 registros obtenidos fueron de los alumnos que se consideraron *usuarios potenciales* del servicio; a estos registros se les aplicó un proceso de conversión de CP a coordenada geográfica, utilizando el conjunto de datos con la información de correspondencia para cada CP, por lo que al conjunto de datos CSV de la matrícula de alumnos se le añadieron dos columna para almacenar la información de la conversión, siendo nombradas: *LONGITUD* y *LATITUD*. En el caso de no encontrarse algún CP en el archivo con dicha información, se procedió a dejar la *LATITUD* y *LONGITUD* en un valor por omisión de 0.0. El motivo de que algunos CP no pudieron ser convertidos a una coordenada correspondiente, se debe a que algunos CP no fueron encontrados en la base de datos de los *Códigos Postales de México* [61], por esta razón se eliminaron los registros con *LATITUD* y *LONGITUD* igual a 0.0. Con este proceso, los registros pasaron de ser 39,700 a 39,652, teniendo una pérdida de 48 registros, el equivalente a un 0.12%.

Con los registros de alumnos “válidos”, es decir, que no contaban con datos faltantes y que además se encontraban en la zona de servicio propuesta, fue factible su análisis permitiendo asignar un volumen de alumnos para los planteles “B ” de las facultades de *Contaduría y Administración*, así como de *Turismo y Gastronomía*, debido a que la matrícula de alumnos no distinguía a los alumnos que estudiaban en un plantel principal o uno secundario; por lo que se propuso hacer un proceso de distribución equitativo, de tal manera que la distribución total de alumnos por cada Facultad mencionada se distribuyó en un 50% para los planteles A y B (principal y secundario). Mediante el gráfico de barras de la Figura 5.1 se observa la distribución de los alumnos, dentro de la zona de servicio,

que estudian en los 52 EA principales más los dos adicionales; se observó que los EA correspondientes a preparatorias presentan el mayor volumen de alumnos.

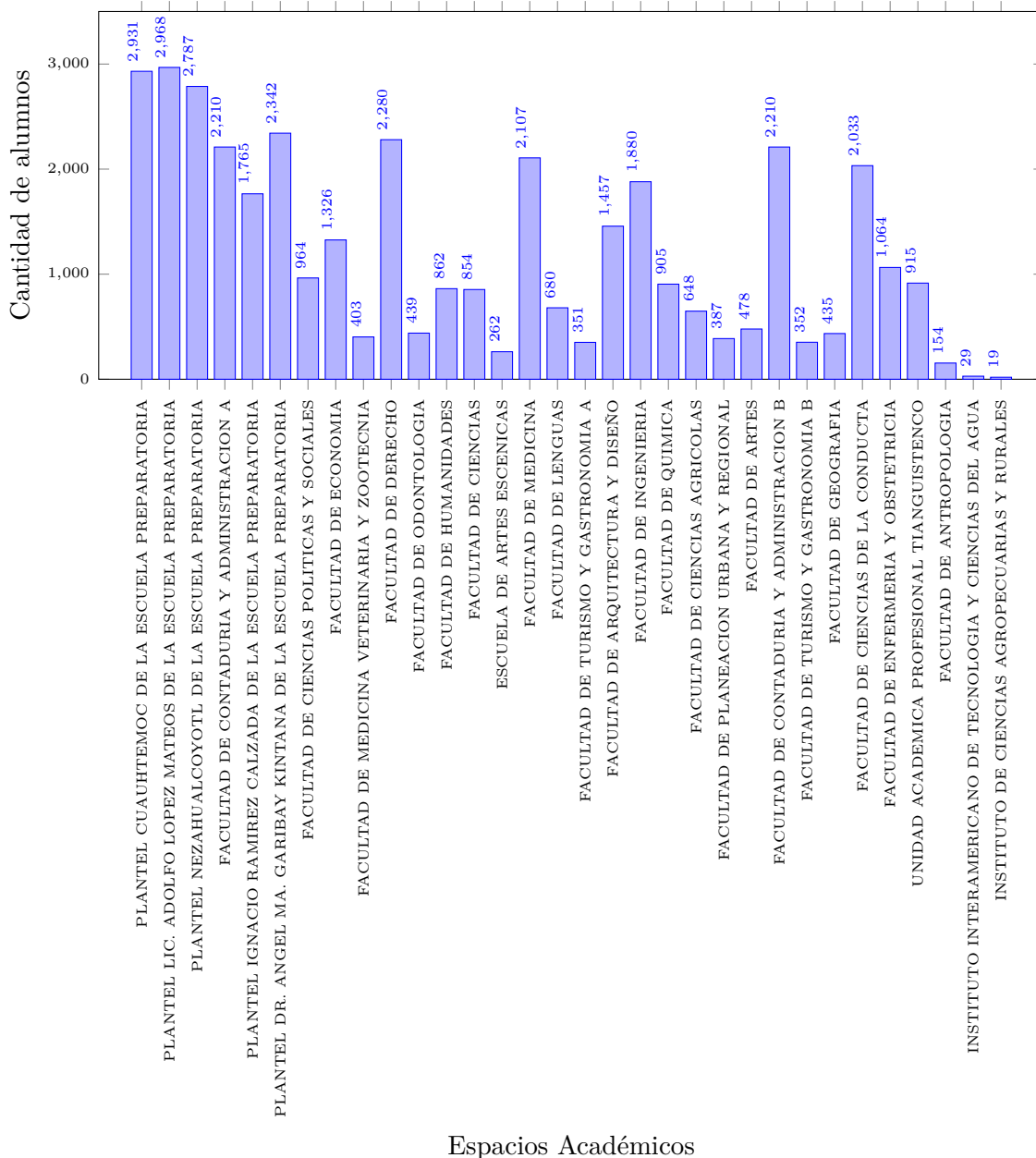


Figura 5.1: Distribución de los alumnos para cada EA dentro de la zona de servicio (Elaboración propia).

## 5.2. Proceso de clustering

Con lo descrito por Sánchez [7], un proceso de clustering puede reducir la cantidad de datos eficientemente, para determinar puntos únicos (clusters) que pueden tener la función de convertirse en *paradas de autobús* y hacer que cada cluster tenga un volumen de alumnos igual a la suma del volumen de estudiantes por cada EA. Pero yendo más allá de este planteamiento, se realizaron varias etapas de clustering con el objetivo de determinar parámetros más específicos para un modelo de ILP. De manera concreta se propuso, mediante el proceso de clustering, determinar del conjunto de datos de la matrícula de alumnos los elementos que pueden tomar los siguientes roles:

- Depósito: Un único punto donde convergen la mayoría de rutas.
- Estaciones: Puntos de ascenso y descenso de alumnos (conformados por EA) y dónde una ruta puede terminar o ser un punto de conexión a otra.
- Paradas de tipo 1: Puntos de ascenso o descenso de alumnos.
- Paradas de tipo 2: Puntos de ascenso de alumnos (ubicación de los grupos de las residencias de los alumnos).
- Terminales: Puntos de paradas más alejados, donde comienzan o finalizan las rutas cuando se realiza la subruta de IDA o la subruta de REGRESO, respectivamente.

En las siguientes secciones se detalla cada uno de los procesos de clustering utilizados para encontrar los elementos descritos.

### 5.2.1. Clustering de Espacios Académicos

Primeramente, al analizar la distribución de los registros de alumnos por EA, se observó que hay registros de alumnos que se dirigían a EA que estaban fuera de la zona de servicio del autobús, además había EA que se encontraban muy cercanos uno de otro, esto se observó al ver sus coordenadas en un plano cartesiano, ver Figura 5.2, por lo que se determinó realizar un proceso de filtrado con los datos de la ubicación de los EA, al igual que en la sección 5.1.2, con lo cual se determinó el conjunto de grupos de EA a donde los alumnos se dirigen y facilitar su análisis, es decir, que se determinaron los puntos de “destino”.

Se aplicaron los mismos pasos de filtrado por CP que se realizaron a la matrícula de alumnos, donde se filtraron aquellos registros por CP de la zona de servicio. Con lo cual los 54 EA se redujeron a 32, por lo que se utilizó el 59.25 % de las Preparatorias y Facultados de la Universidad Autónoma del Estado de México (UAEMex). Teniendo esto, se realizó un proceso de clustering mediante el algoritmo DBSCAN aplicado a los 32 registros de los EA determinados. Este proceso de clustering utilizó solamente dos características: LONGITUD y LATITUD, por lo que fue factible utilizar la métrica de distancia *Haversine*,

que es adecuada para coordenadas geográficas, por lo que se definió una distancia para el parámetro *EPS* en kilómetros; este parámetro de DBSCAN se dejó en un valor de 1.0 que expresa 1 km de distancia.

El proceso de clustering añadió una etiqueta numérica que representó el grupo al cual cada EA fue asignado por DBSCAN, es decir, se añadió un identificador de cluster. Al haber utilizado DBSCAN y establecer su parámetro *MINPTS* en 1, se omitió la generación de elementos no agrupados (outliers), por lo que el resultado del agrupamiento de los 32 EA fue de 11 clusters etiquetados de 0 a 10.

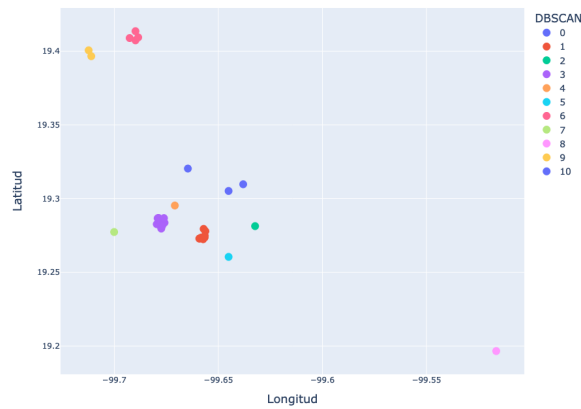
Teniendo los grupos formados por la primera etapa de clustering de los EA, se procedió a hacer la correspondencia con los registros en la matrícula de alumnos añadiendo una columna denominada *GRUPO\_EA*, donde se guardó el valor del identificador del cluster que corresponde a los grupos de EA formados anteriormente. En este proceso se añadió un valor adicional al etiquetado de los clusters, siendo el valor de  $-1$ , con el objetivo de identificar aquellos registros de alumnos que están en la zona de servicio, pero estudian en algún EA que no está en dicha zona y por ende su EA correspondiente no se encontró en el grupo de EA determinado por el clustering. En la matrícula de alumnos se eliminaron los registros con el campo *GRUPO\_EA* con valor de  $-1$ , ya que estos registros contenían a los alumnos que se dirigen a los EA que no estaban en la zona de servicio del potrobús. Al realizar esta operación el conjunto de datos pasó de 39,652 a 38,496, con lo cual se omitieron un total de 1,157 registros, el equivalente a un 2.9 %.

Al haber realizado un proceso de clustering con los EA, se pudo determinar la distribución de alumnos por grupo; este punto fue crucial ya que estos grupos fueron catalogados como: Depósitos, Estaciones o Paradas, dependiendo del volumen de alumnos y cantidad de EA que contenía cada grupo. En la Tabla 5.3 se muestra el ID del cluster determinado por DBSCAN, así como los EA que conformaban a cada cluster y su respectivo volumen de alumnos.

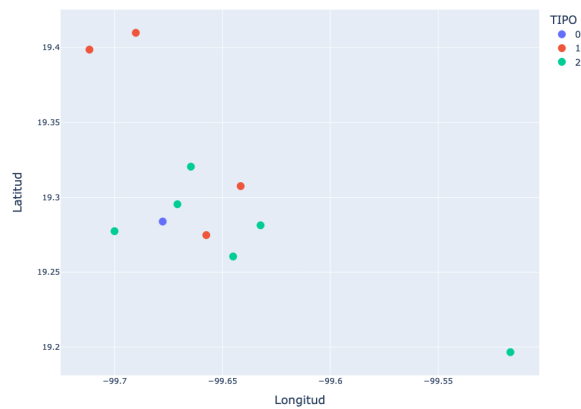
Tabla 5.3: Clusters formados de los Espacios Académicos (Elaboración propia).

ID Cluster	Espacios académicos	Volumen de alumnos
0	Fac. de Ciencias de la Conducta y la Preparatoria #3	4,964
1	Campus Colón	8,704
2	Preparatoria #2 - Nezahualcoyotl	2,787
3	Campus CU	12,244
4	Preparatoria #4 - Ignacio Ramirez Calzada	1,765
5	Preparatoria #5 - Angel Maria Garibay	2,342
6	Campus El Cerrillo	1,924
7	Escuela de Artes Escénicas	262
8	Unidad Académica Profesional de Santiago Tianguistenco	915
9	IITCA y la Fac. de Gastronomía campus B	380
10	Fac. de Contaduría campus B	2,209

Con el volumen de alumnos por cada grupo de EA y utilizando la cantidad de EA por grupo se determinó un campo denominado TIPO. Se estableció una jerarquía en cada grupo de EA con la finalidad de formar subconjuntos de puntos y facilitar la formulación del modelo matemático descrito más a detalle en la sección 5.3. Con este criterio se determinó que el cluster 3 fuese el depósito ( $TIPO = 0$ ), mientras que los clusters 0, 1, 6, 9 se designaron como estaciones ( $TIPO = 1$ ) y finalmente los clusters restantes 2, 4, 5, 7, 8 y 10 se designaron como paradas de espacios académicos ( $TIPO = 2$ ). En los gráficos de la Figura 5.2 se observa la formación de grupos de EA, en la Figura 5.2a se aprecia la ubicación de los puntos en un plano, por otro lado los puntos mostrados en la Figura 5.2b expresan los centroides de los EA por grupo.



(a) Ubicación EA con la distinción de los clusters formados.



(b) Ubicación de los Grupos de EA con el cálculo del centroide de cada EA por grupo.

Figura 5.2: Proceso de clustering de los EA (Elaboración propia).

## Simulación de horarios

Los registros de alumnos en la matrícula no contaban con algún dato adecuado para ser consultados por horarios, así que se añadió esta característica mediante la asignación de horarios aleatorios a los registros para simular tanto los horarios de entrada, así como los de salida de cada alumno. Se propuso lo siguiente para las preparatorias de la UAEMex: “se sabe que los alumnos de bachillerato tienen dos horarios (matutino y vespertino), por lo cual es conveniente asignar horarios de este tipo a los 12,793 alumnos que estudian en este nivel”. En el caso de los horarios para el nivel de Licenciatura, al ser los horarios más variados se estipularon en intervalos de dos horas, esto se observa en la Tabla 5.1.

Después del proceso de filtrado y los pasos necesarios para completar el conjunto de datos de la matrícula, los 38,496 registros representan a los alumnos que se encontraban en la zona de servicio y estudiaban en EA dentro de la misma zona y por consiguiente son a los cuales se les pudo ofertar el servicio de transporte. A este volumen de datos se les aplicó un proceso de simulación de horarios, con el fin de analizar el comportamiento de los datos y aplicar el proceso de clustering teniendo un aproximado de los alumnos que ocuparían el servicio de transporte en ciertas horas.

## Determinación del mejor valor EPS para el clustering de la matrícula de alumnos

El proceso de clustering para la matrícula de alumnos se realizó utilizando tres características que presentaban los datos para poder ser agrupados: LONGITUD, LATITUD y GRUPO\_EA. Estas tres características fueron determinadas y completadas mediante el preprocesamiento de datos mostrado en la sección 5.1. Para poder utilizar estas tres características fue necesario una métrica de distancia que permitiese medir en más de dos dimensiones, por lo que se utilizó la métrica *euclidiana*, debido a su fácil correspondencia a km, siendo 1 km equivalente a 0.01. Si bien esta configuración parecía ideal, fue necesario valorar la calidad de los clusters obtenidos, para decidir un valor adecuado para el parámetro *EPS* [7].

En [37] se propuso una distancia adecuada que podría recorrer un estudiante para trasladarse a una parada cercana a su domicilio, siendo el valor de 1.5 km establecida como adecuada. Para el proceso de pruebas con EPS, se decidió establecer un conjunto de valores: 1 km, 1.25 km, 1.5 km, 1.75 km y 2 km, con su equivalencia a una métrica euclidiana, siendo los valores de 0.01, 0.012, 0.015, 0.02 respectivamente.

La valoración de la calidad del proceso de clustering realizado, dependió de la calidad misma de los clusters formados y con el objetivo de determinar el mejor valor para el parámetro EPS, se utilizaron las siguientes métricas internas de calidad mencionadas en [7]: *Silhouette Score*, *Índice Davies Bouldin*, *Índice Calinski Harabasz*.

Con estas métricas internas de valoración, se asegura que el parámetro *EPS* del algoritmo DBSCAN sea seleccionado de manera adecuada basado en los criterios de: distancia, similaridad y cohesión.

Las expresiones matemáticas para cada métrica mencionadas, así como la interpretación de sus resultados son descritas en la sección 2.1.3. En la Tabla 5.4 se muestran los resultados con variaciones al parámetro *EPS*, donde el valor EPS en 0.01 obtuvo mejores puntajes para las métricas Silhouette Score y Calinski-Harabasz.

Tabla 5.4: Resultados de las métricas de valoración de calidad de clusters (Elaboración propia).

EPS	Silhouette Score	Índice Davies-Bouldin	Índice Calinski-Harabasz
<b>0.01</b>	<b>0.7829</b>	1.5253	<b>2576.6605</b>
0.0125	0.4644	1.4849	1582.9912
0.015	0.4673	1.4846	1736.0836
0.0175	0.4634	1.4726	1930.4249
0.02	0.3781	<b>1.4379</b>	1569.9588

Sin embargo, las tres métricas internas utilizadas pertenecen a escalas diferentes, y para evitar una selección subjetiva (selección del EPS basado en la elección del mejor de dos de tres pruebas), se aplicó un proceso de Análisis de Decisiones Multicriterio (MCDA, por sus siglas en inglés) [62]. Este proceso consistió en una conversión de las métricas (cada una con escalas diferentes) a una escala común (normalización), en este caso para una escala comprendida entre  $[0, 1]$ ; posteriormente estas métricas fueron combinadas para obtener un *Índice General de Calidad* expresado como:  $I(EPs)$ .

Para las métricas Silhouette Score y Calinski-Harabasz, cuyos valores altos se interpreta como una mejor calidad de los clusters, la Ecuación 5.1 expresa la normalización aplicada para estas métricas; dónde  $X_{norm}$  es el valor normalizado,  $X_{min}$  y  $X_{max}$  representan los valores mínimos y máximos de la métrica y finalmente  $X$  es el valor a normalizar.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (5.1)$$

En el caso de la métrica *David-Bouldin*, cuyos valores bajos expresan mejor calidad de los clusters, la Ecuación 5.2 debe implementarse para el proceso de normalización.

$$X_{norm} = \frac{X_{max} - X}{X_{max} - X_{min}} \quad (5.2)$$

Con estos procesos se aseguró que los valores quedaran entre 0 y 1, dónde un valor de 1 expresa el valor óptimo de calidad de clusters. El cálculo de  $I(EPs)$  se realizó mediante la Ecuación 5.3, en donde  $w_1$ ,  $w_2$  y  $w_3$  representan las ponderaciones para cada una de las métricas.

$$I(EPs) = w_1S(EPs) + w_2C(EPs) + w_3D(EPs) \quad (5.3)$$

donde:

- $S$  = Silhouette Score normalizado.
- $C$  = Calinski-Harabasz normalizado.
- $D$  = Davies-Bouldin normalizado.
- $w_1, w_2, w_3$  = ponderaciones para priorizar alguna de las métricas.

Debido a que los pesos para las tres métricas se consideraron bajo la misma prioridad, es decir,  $w_1 = w_2 = w_3 = \frac{1}{3}$ ; la Ecuación 5.3 para el cálculo del Índice General de Calidad se puede expresar mediante la Ecuación 5.4.

$$I(EPs) = \frac{S + C + D}{3} \quad (5.4)$$

En la Tabla 5.5 se observan los resultados obtenidos de las métricas ya normalizadas, así como el valor calculado del “Índice General de Calidad”; con lo cual se puede concluir que el mejor valor para el parámetro  $EPs$  para el algoritmo DBSCAN es 0.01 con un índice de calidad general  $I(EPs)$  de 0.6667.

Tabla 5.5: Resultados de las métricas de valoración de calidad de clusters con normalización(Elaboración propia).

<b>EPS</b>	<b>Silhouette Score Normalizado</b>	<b>Índice Davies-Bouldin Normalizado</b>	<b>Índice Calinski-Harabasz Normalizado</b>	<b><math>I(EPs)</math></b>
<b>0.01</b>	1.0	0.0	1.0	<b>0.6667</b>
0.0125	0.2132	0.4627	0.0129	0.2296
0.015	0.2204	0.4662	0.1650	0.2839
0.0175	0.2107	0.6036	0.3581	0.3908
0.02	0.0	1.0	0.0	0.3333

A diferencia de otros algoritmos de clustering como K-means, DBSCAN tienen la característica de determinar elementos outliers (demasiado alejados a otros puntos para ser considerados clusters) [7]. Mediante el gráfico de barras de la Figura 5.3 se observa tanto la cantidad de clusters, así como los elementos outliers para cada valor  $EPs$ .

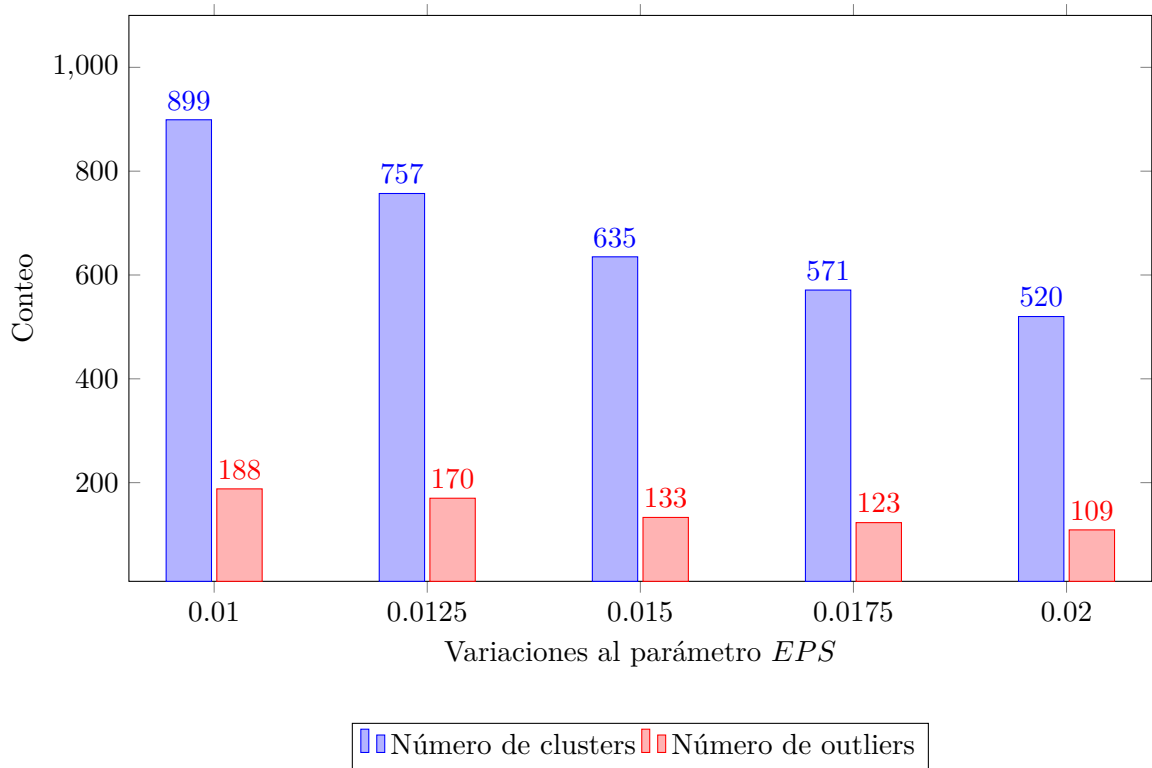


Figura 5.3: Conteo de clusters y outliers al variar el parámetro *EPS* (Elaboración propia).

### 5.2.2. Proceso de clustering de la matrícula de alumnos

Según las métricas implementadas, el mejor valor *EPS*, para tener una buena calidad de los clusters formados, fue de 0.01 (equivalente a una distancia de 1 km), con lo cual se procedió a realizar el proceso de clustering con dicho valor. Primeramente se realizó éste proceso con todo el volumen de datos de la etapa anterior de preprocesamiento, es decir, sin un filtrado por horario, con el objetivo de hacer un análisis general de la distribución de los puntos, la cantidad de grupos y elementos *outliers* formados. Mediante el gráfico mostrado en la Figura 5.4 se aprecia la distribución de los grupos formados por la matrícula de alumnos en la zona de servicio.

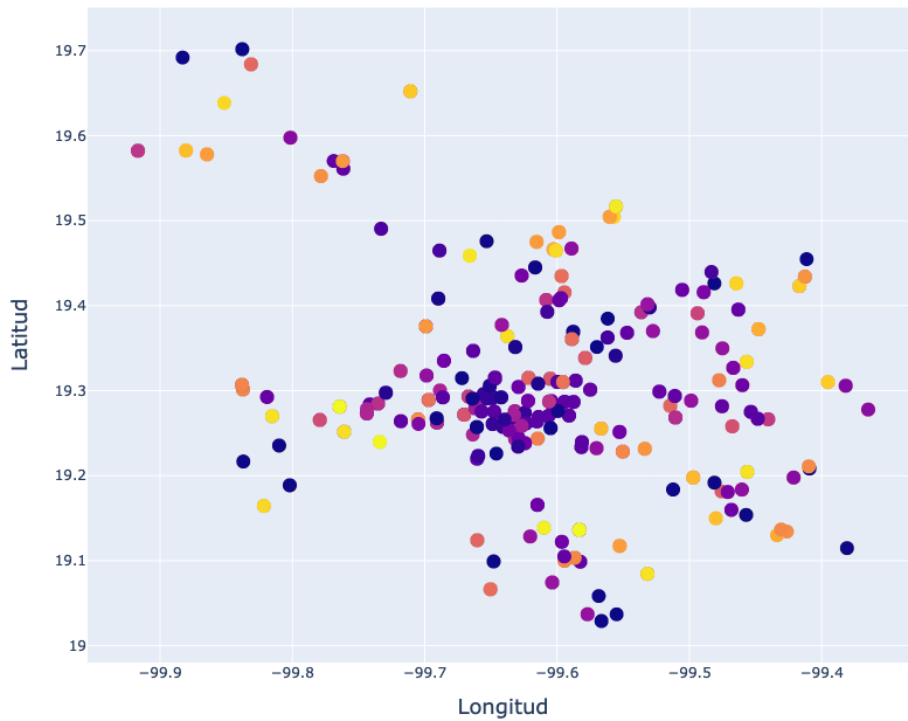
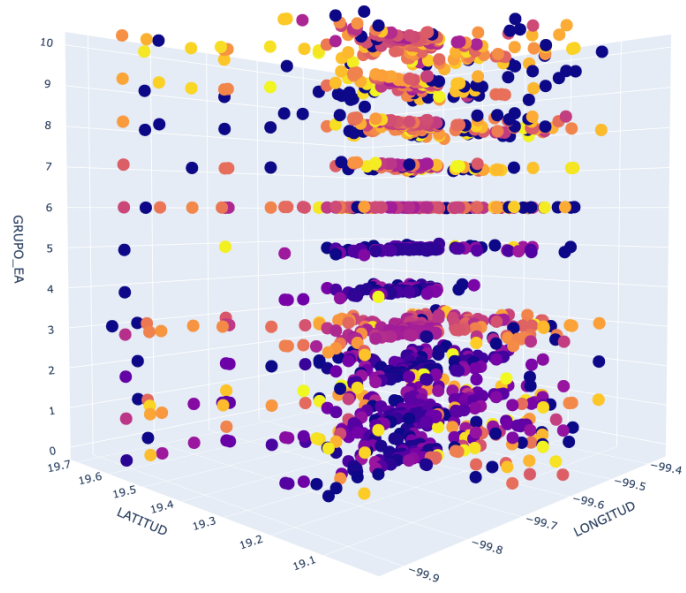
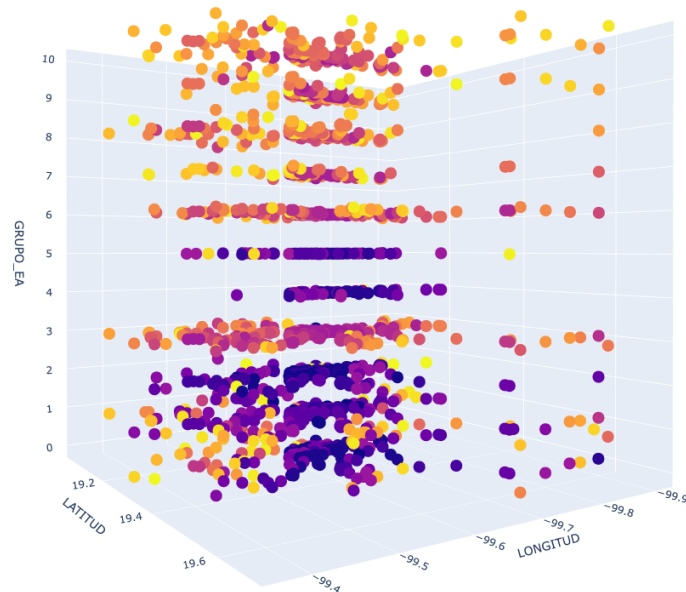


Figura 5.4: Resultados del proceso de clustering en un plano de dos dimensiones (Elaboración propia).

Al haber utilizado tres características ( $X$ ,  $Y$  y  $Z$ ) en el proceso de clustering, correspondientes a la LONGITUD, LATITUD Y GRUPO\_EA, fue posible hacer un análisis gráfico de los resultados en un plano de tres dimensiones. En el gráfico de la Figura 5.5a se puede observar los resultados del proceso de clustering de los 38,496 registros de alumnos. Con ese gráfico podemos determinar *capas*, que representan la distribución de los alumnos hacia los 11 Grupos de EA, obtenidos anteriormente, identificados numéricamente del 0 al 10 en el eje GRUPO\_EA. Mientras que con el gráfico de la Figura 5.5b se observa la distribución de clusters pero sin elementos outliers.



(a) Clusters formados.



(b) Clusters formados sin outliers.

Figura 5.5: Proceso de clustering con tres características (Elaboración propia).

### 5.2.3. Proceso de clustering de la matrícula de alumnos con distinción de horarios

Con los resultados del proceso de clustering de toda la matrícula de alumnos, se planteó realizar el mismo proceso de clustering pero con un conjunto de datos en un horario determinado, en este caso a las 7:00 am. Mediante este criterio se observó que de la matrícula de alumnos se tenían 9,968 registros de alumnos que ingresaban a esa hora a sus EA. Se determinó que los alumnos se concentran más en el GRUPO\_EA = 1, que representa al campus de Colón. Sin embargo no fue viable utilizar el criterio del volumen de alumnos para establecer a este grupo como el *Depósito o Campus Principal*, ya que este grupo fue muy cambiante en cuanto a su volumen de alumnos, debido a que se analizó la distribución de alumnos en otro horario distinto a las 7:00 am, como es el caso de las 9:00 am, el volumen de alumnos bajó drásticamente, dado a que en los horarios de entrada de los alumnos de Licenciatura coincidían con los de nivel Bachillerato; esto puede ser observado con el gráfico de la Figura 5.6. Con el análisis realizado de las 7:00 am, siendo este el momento de *máxima afluencia de alumnos*, se pudo tomar este horario como un escenario para el análisis del volumen de alumnos por capa.

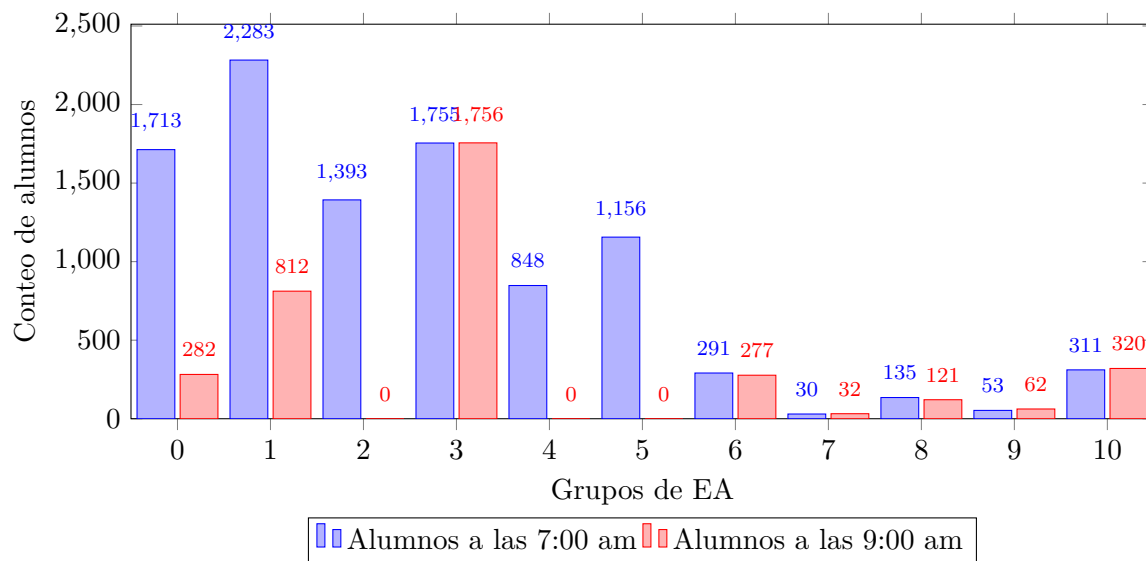
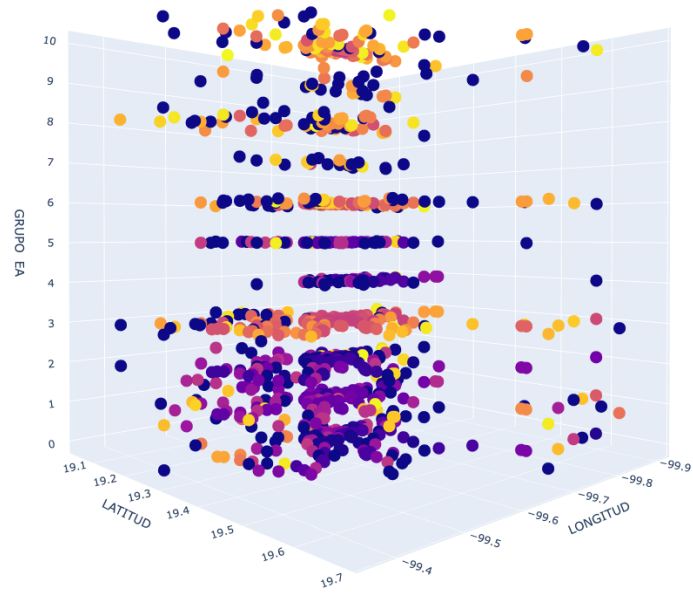
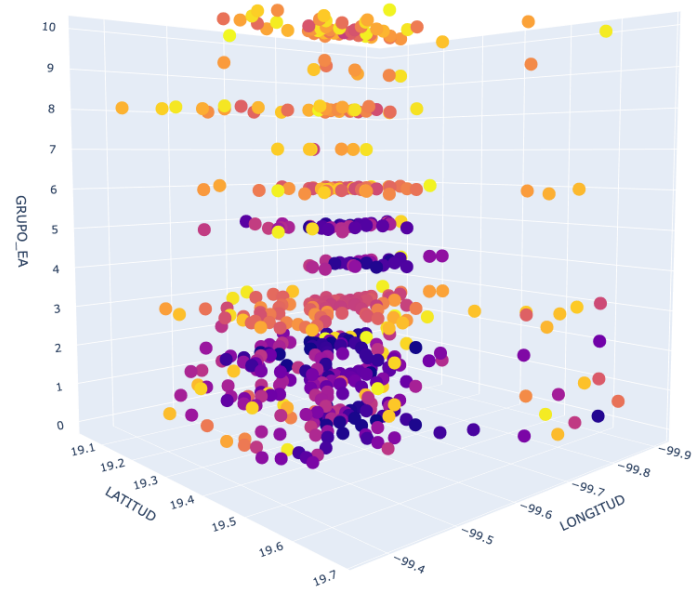


Figura 5.6: Volumen de alumnos en dos horarios distintos (Elaboración propia).

En los gráficos de la Figura 5.7, se aprecia la distribución de alumnos en un plano de tres dimensiones tanto con elementos outliers, ver Figura 5.7a, como sin estos, ver Figura 5.7b. Con estos resultados y lo establecido en la sección 5.2.1, se generó la primera parte de los datos de salida con la información de demanda de alumnos por cada grupos de EA del primer proceso de clustering.



(a) Clusters formados.



(b) Clusters formados sin outliers.

Figura 5.7: Proceso de clustering con tres características en un horario de 7:00 am (Elaboración propia).

#### 5.2.4. Análisis por capas en un horario determinado

Las particularidades de la distribución de alumnos hacia alguno de los 11 destinos definidos en la sección 5.2.1, permitieron hacer un análisis por capa del segundo proceso de clustering; donde se analizó cada una de estas capas para encontrar los puntos de las ubicaciones de los alumnos cercanos a sus respectivos EA, los cuales pudieron ser descartados de los datos finales para el modelo matemático de ILP, mostrado en la sección 5.3. En las imágenes de la Figura 5.8 se muestra el análisis de los puntos en la capa 3 (Campus CU), los puntos de tonalidad azul en la Figura 5.8a corresponden a las ubicaciones de los alumnos clusterizados que estudian en los EA agrupados en el  $\text{GRUPO\_EA} = 3$ , mientras que el punto rojo expresa la ubicación de ese grupo de EA. En la Figura 5.8b se observa una circunferencia de un radio de 2 km cuyo centro es el Grupo de EA correspondiente a esa capa, los puntos azules dentro de esta circunferencia se descartaron como posibles paradas por cercanía a su respectivo campus (punto rojo), ya que a esa distancia los alumnos se dirigen a su EA caminado o en otro tipo de transporte personal.

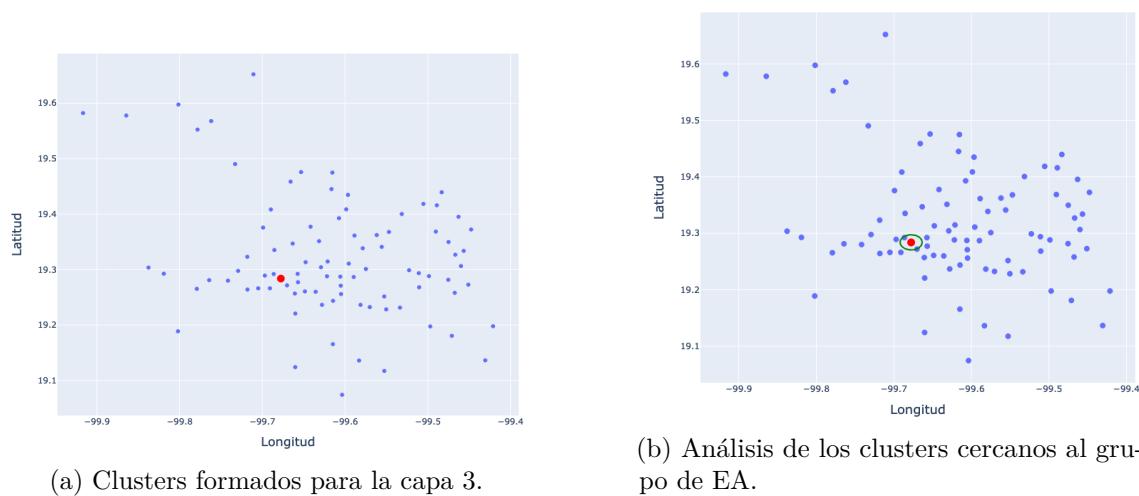


Figura 5.8: Análisis de la capa 3 - Campus CU (Elaboración propia).

De igual manera para la capa 3, en la Figura 5.9 se muestra un análisis mediante un gráfico con KDE, descrito en la sección 2.1.5, con el cual se observó la concentración de densidad (volumen de alumnos) cercana a su respectivo campus. Con este análisis se puede corroborar que hay una cantidad considerable de alumnos cerca del destino (campus) donde ellos se dirigen.

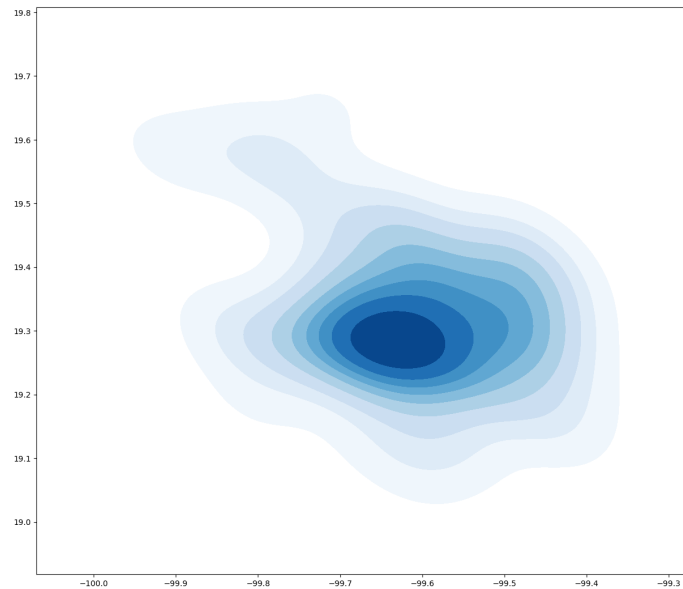


Figura 5.9: Análisis mediante KDE para la capa 3 (Elaboración propia).

### 5.2.5. Definición de los datos de salida para “alimentar” el modelo matemático de ILP

Teniendo la distribución de alumnos por Grupo de EA, se procedió a generar el compendio de datos resultantes de los procesos de clustering, para esto se requirió crear la primera parte de este conjunto de datos, que solo se enfocó en la determinación de Depósito, Estaciones y EAs que se consideraron Paradas. Este nuevo conjunto de datos se conformó por los campos de:

- ID\_GRUPO: Identificador para distinguir el grupo que representa (del 0 al 10)
- LONGITUD: Coordenada en posición x
- LATITUD: Coordenada en posición y
- TIPO: Campo que almacena el tipo de vértice, siendo 0 el depósito, 1 estaciones, 2 paradas de EA, 3 paradas de clusters de alumnos y 4 siendo las Terminales.
- VOLUMEN\_ALUMNOS: Conteo de la cantidad de alumnos que pueden llegar a este grupo.
- ELEMENTOS: Descriptor que enlista los elementos que compone al grupo de datos, en este caso cada uno de los EA que forman parte de este grupo.

### Análisis de los clusters formados en un horario determinado

Con el conteo de registros de los alumnos por grupo formado con el proceso de clustering, se pudo observar que en este segundo proceso de clustering, la cantidad de alumnos que se distribuyó en grupos denominados outliers (valor de  $-1$ ) fue de 216, teniendo una cantidad de 526 clusters formados. Cada uno de estos grupos obtenidos por el clustering de DBSCAN se les calculó su centroide, además se les asignó un valor numérico de 3 en la columna TIPO para diferenciarlos de las paradas obtenidas en la sección 5.2.1 y poder determinar el subconjunto de vértices “paradas” formados por el clustering DBSCAN en los registros de alumnos. En la Figura 5.10 se muestran los puntos únicos que representa los grupos determinados por DBSCAN mediante el cálculo de su centroide omitiendo los outliers.

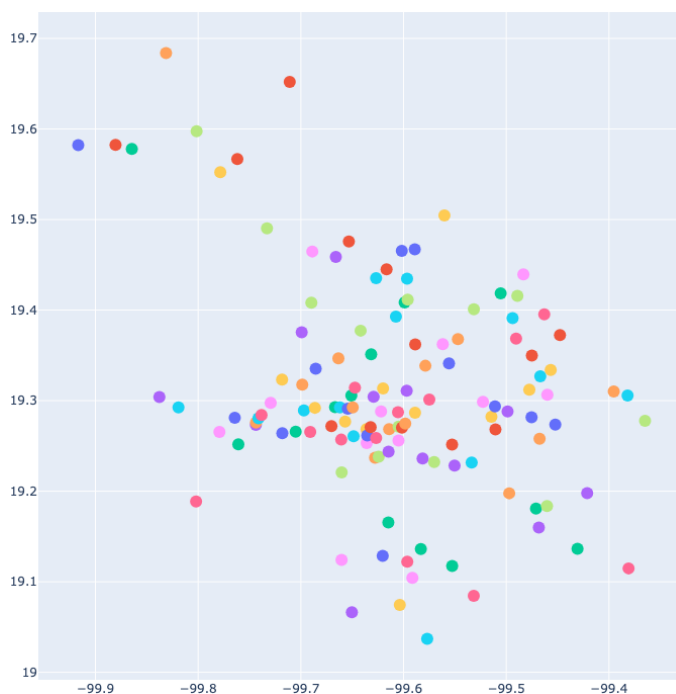


Figura 5.10: Puntos determinados por los procesos de clustering (Elaboración propia).

Si bien, el proceso de clustering a tres dimensiones nos permitió identificar el grupo de alumnos que estudian en uno de los 11 Grupos de EAs, está presente la siguiente problemática: *Existen grupos de alumnos que se ubican en la misma posición  $X$ ,  $Y$  (longitud y latitud), pero se distribuyen a destinos diferentes.* Esta problemática ocasionó que varios puntos se encimaran, ver Figura 5.10, por lo que se propuso hacer una combinación de estos elementos sin perder su distribución y con el único objetivo de reducir lo más posible la cantidad de puntos para el trazado de las rutas por el modelo matemático.

La combinación fue a cargo de un tercer proceso de clustering mediante DBSCAN usando solamente dos características: *longitud* y *latitud* utilizando un valor bajo para el parámetro EPS, se propuso una distancia de 250 metros (equivalente a 0.0025). Mediante este proceso se agruparon los “grupos cercanos” a las ubicaciones de los alumnos, logrando identificar un solo punto para representar hasta los 11 destinos a los que se dirige un grupo de alumnos en dicho punto. Con este proceso y añadiendo columnas adicionales a los datos, para evitar perder información, se redujo el número de puntos a una cantidad de 128; los cuales se establecieron como *paradas* de *TIPO* = 3.

Con las tres etapas de clustering se consiguió obtener un número adecuado de puntos de parada, siendo un total de 128, retomando lo mencionado en la sección 2.1.1 sobre la complejidad del algoritmo de clustering DBSCAN y dado a que se utilizó una matriz de distancias cuadrada, la complejidad de este algoritmo, aplicada a este caso de estudio, es de  $O(n^2)$ . Sin embargo, a pesar de que la complejidad de DBSCAN se encuentra en “el peor de los casos” su aplicación práctica es factible. En la Tabla 5.6 se puede observar los tiempos tomados en segundos para la ejecución de las tres etapas de los procesos de clustering con DBSCAN.

Tabla 5.6: Resultados de los tiempos de ejecución de las tres etapas de DBSCAN (Elaboración propia).

<b>Etapas de DBSCAN</b>	<b>Número de datos</b>	<b>Métrica</b>	<b>Dimensiones</b>	<b>Tiempo de CPU</b>
Clustering de EA	32	Haversine	2	0.044 seg
Clustering de Alumnos	9,968	Euclidiana	3	0.285 seg
Clustering de Eliminación	526	Euclidiana	2	0.012 seg

A los 128 puntos determinados por los procesos de clustering se les aplicó una *Envolvente Convexa*, ver Figura 5.11, con el fin de determinar los puntos a ser considerados *Terminales*. En esta etapa se determinó un total de 9 puntos terminales, de los cuales debe salir al menos una ruta por cada elemento en este subconjunto para el modelo de ILP. Estos puntos terminales se les asignó el valor de 4 en el campo de *TIPO* en el conjunto de datos de salida.

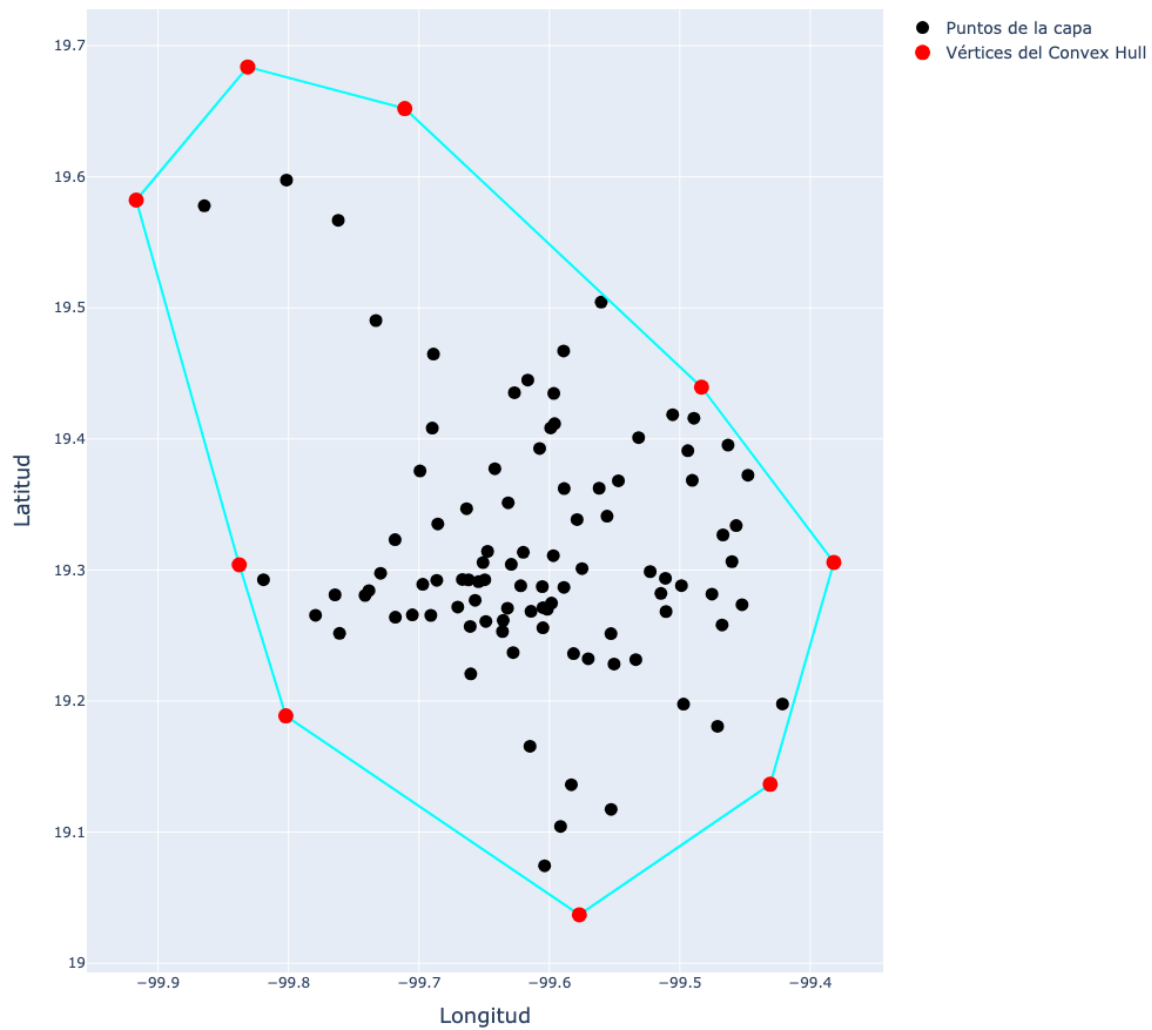


Figura 5.11: Implementación de la Envoltente Convexa (Elaboración propia).

En el gráfico mostrada en la Figura 5.12 se puede observar los puntos finales ya catalogados, en donde el depósito está marcado como  $TIPO = 0$ , las estaciones tienen  $TIPO = 1$ , las paradas de EA tienen  $TIPO = 2$ , las paradas de la ubicación de los alumnos tienen  $TIPO = 3$  y las terminales tienen  $TIPO = 4$ .



Figura 5.12: Puntos finales para el modelo ILP (Elaboración propia).

### 5.3. Modelos de ILP para el caso de estudio

Con lo mencionado en la sección 5.2, un proceso de clustering con tres características permitió determinar el volumen de alumnos que se dirigen a un grupo de EA en particular, interpretado como *capas*, la cantidad total de capas dependió de la totalidad de grupos de EA que se determinaron, por lo que se establecieron un total de 11. Teniendo construido el conjunto de datos, por los procesos de clustering, mediante los componentes: *Depósito*, *Estaciones*, *Paradas tipo 1*, *Paradas tipo 2* (EA sin agrupar y paradas definidas por los clusters respectivamente) y un conjunto de *Terminales*, se procedió a construir el modelo matemático de ILP.

En esta sección se presentan dos propuestas de modelo de ILP para el caso de estudio; siendo un modelo basado en un CVRP con una variable de decisión principal con dos índices como primera alternativa y como segunda alternativa se presenta un modelo de enrutamiento basado en VRP en combinación con el problema de flujo máximo para solventar las necesidades de flujo del servicio de transporte.

### 5.3.1. Consideraciones generales para los modelos de ILP

Los aspectos generales considerados para los modelos presentados en las secciones 5.3.2 y 5.3.3 son: se tienen un flota de vehículos con una capacidad  $C$  homogénea, al conjunto conformado por depósito, estaciones deben llegar al menos a una ruta pero sin retornar a los puntos de inicio, hay un subconjunto de vértices terminales de donde una ruta inicia  $F$  (uno para cada ruta), se deben satisfacer un conjunto de peticiones de transporte: flota de vehículos (autobuses) disponibles, cumplir con horarios de llegada y salida determinados en cada vértice.

Para ambos modelos se consideraron los mismos subconjuntos de vértices, con algunas variaciones mínimas, por lo que siendo  $G = (V, A)$  una gráfica con  $V$  como el conjunto de vértices y  $A$  como el conjunto de aristas o arcos; del conjunto de vértices  $V = \{0, 1, \dots, n-1\}$ , se denota al vértice  $\{0\}$  como el depósito,  $E$  el subconjunto de vértices denominado estaciones,  $S1$  el subconjunto de paradas tipo 1,  $S2$  el subconjunto de paradas tipo 2 y siendo  $F$  el subconjunto de vértices terminales, de tal manera que  $V = \{0\} \cup E \cup S1 \cup S2 \cup F$ . Existe un costo  $c_{ij}$  que expresa el tiempo de ir desde el vértice  $i$  al  $j$ , calculado mediante la consideración de una velocidad promedio para cada vehículo de 40 km/hr. Para cada vértice  $i \in V$  se tienen un volumen de alumnos total denotado por  $B_i$  y para cada vértice  $i$  su volumen total de alumnos se distribuye en los 11 destinos, denotado por  $b_{id}$  por lo que existe un subconjunto de vértices destinos  $D = \{0\} \cup E \cup S1$ . Los modelos presentados se enfocan en la construcción de rutas en sentido de IDA, sin embargo, también se describe el proceso para establecer un modelado de las rutas en sentido de REGRESO.

### 5.3.2. Modelo a dos índices

Este modelo de ILP se formuló a dos índices tomando de base el CVRP, por lo que se definió una variable de decisión binaria  $x_{ij}$  que toma el valor de 1 si la arista de  $i$  a  $j$  es utilizado, 0 en caso contrario. Para modelar la capacidad de pasajeros se tienen una variable  $Q_i$  que expresa la cantidad de pasajeros que se tienen al llegar a la parada  $i$ . Se incluyó variables para darle características de flujo máximo, siendo  $z_i$  la variable entera que determina la cantidad de aristas entrantes y salientes por vértice, además se incluye una variable entera  $y_{ij}$  para modelar el flujo por la arista  $i$  a  $j$ . Se define además una constante  $R$  para establecer el máximo número de aristas entrantes a un vértice siendo  $R = |F|$ .

El modelo matemático a dos índices para la IDA se muestra a continuación:

$$\text{mín} \sum_{i \in V} \sum_{j \in V, i \neq j} c_{ij} x_{ij} \quad (5.5)$$

$$z_0 = R \quad (5.6)$$

$$z_i = \left\lceil \frac{B_i}{C} \right\rceil \quad \forall i \in V \setminus \{0\} \quad (5.7)$$

$$\sum_{j \in V \setminus F} x_{ij} = z_i \quad \forall i \in F \quad (5.8)$$

$$\sum_{j \in V} x_{ji} = 0 \quad \forall i \in F \quad (5.9)$$

$$\sum_{j \in V \setminus F} x_{ij} = z_i \quad \forall i \in S2 \quad (5.10)$$

$$\sum_{j \in V} x_{ji} = z_i \quad \forall i \in S2 \quad (5.11)$$

$$\sum_{j \in V \setminus F} x_{ij} = z_i \quad \forall i \in S1 \quad (5.12)$$

$$\sum_{j \in V} x_{ji} = z_i \quad \forall i \in S1 \quad (5.13)$$

$$z_i \leq \sum_{j \in V \setminus F} x_{ij} \leq R \quad \forall i \in E \quad (5.14)$$

$$z_i \leq \sum_{j \in V} x_{ji} \leq R \quad \forall i \in E \quad (5.15)$$

$$0 \leq \sum_{j \in V \setminus (\{0\} \cup F)} x_{0j} \leq z_0 \quad (5.16)$$

$$0 \leq \sum_{j \in V \setminus \{0\}} x_{j0} \leq z_0 \quad (5.17)$$

$$\sum_{j \in V} y_{ij} \leq z_i * C \quad \forall i \in V, i \neq j \quad (5.18)$$

$$\sum_{j \in V} y_{ji} = 0 \quad \forall i \in V \setminus (\{0\} \cup F), i \neq j \quad (5.19)$$

$$\sum_{j \in V} y_{ji} = 0 \quad \forall i \in F, i \neq j \quad (5.20)$$

$$x_{ij} = 0 \quad \forall i, j \in V, i = j \quad (5.21)$$

$$Q_i - Q_j + C * x_{ij} \leq C - B_j \quad \forall i, j \in V, i \neq j \quad (5.22)$$

$$B_i \leq Q_i \leq C \quad \forall i \in V \quad (5.23)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in V \quad (5.24)$$

$$y_{ij} \in \mathbb{Z}^+, \quad 1 \leq y_{ij} \leq C \quad \forall i \in V, j \in V \quad (5.25)$$

$$z_i \in \mathbb{Z}^+, \quad 0 \leq z_i \leq R \quad \forall i \in V \quad (5.26)$$

$$Q_i \in \mathbb{Z}^+, \quad Q_i \geq 0 \quad \forall i \in V \quad (5.27)$$

La Ecuación 5.5 establece la función objetivo que busca encontrar el costo mínimo; las Ecuaciones 5.6 y 5.7 permiten determinar el número de aristas entrantes y salientes

para el depósito y los demás vértices; las Ecuaciones 5.9, 5.13, 5.11, 5.15, 5.17 establecen las aristas entrantes para las *terminales*, *paradas tipo 1*, *paradas tipo 2*, *estaciones* y el *depósito* respectivamente; las Ecuaciones 5.8, 5.12, 5.10, 5.14, 5.16 establecen las aristas salientes; la Ecuación 5.16 y 5.17 especifica que en el depósito llega un número de rutas; las Ecuaciones 5.18, 5.19 y 5.20 establecen el flujo máximo entre las aristas salientes y entrantes; la Ecuación 5.21 se utiliza para la eliminación de *loops*; las Ecuaciones 5.22 y 5.23, se encargan de modelar la carga acumulada de cada ruta y la eliminación de subciclos; en las Ecuaciones 5.24 a la 5.27 se definen las variables del modelo.

### 5.3.3. Modelo de VRP con flujo máximo

La formulación matemática para el modelo del caso de estudio utiliza tanto elementos del VRP, así como de flujo máximo para garantizar que se lleve la mayor cantidad de estudiantes hacia su respectivo espacio académico. El modelo se describe mediante una gráfica  $G = (V, A)$ , con  $V$  como el conjunto de vértices y  $A$  como el conjunto de aristas; del conjunto de vértices  $V = \{0, 1, \dots, n - 1\}$  el vértice  $\{0\}$  representa al depósito (campus principal), del conjunto de  $V$  se denota a  $E$  como el subconjunto de vértices denominado estaciones,  $S$  el subconjunto de paradas y siendo  $F$  el subconjunto de vértices establecidos como terminales, de tal manera que  $V = \{0\} \cup E \cup S \cup F$ . Para el flujo máximo se define al subconjunto de destinos  $D = \{0\} \cup E$ . Existe un costo  $c_{ij}$ , que expresa el tiempo de ir desde el vértice  $i$  al  $j$  calculado mediante una velocidad promedio de 40 km/hr; para cada vértice  $d \in D$  se tienen una demanda de alumnos denotado como  $B_d$  y para cada vértice  $i \in S \cup F$  existe un suministro de alumnos que se dirige a un destino  $d \in D$ , expresado como  $b_{id}$ . El conjunto de vehículos se denota como  $K$ , del conjunto de vehículos se tienen dos subconjuntos  $K_P$  y  $K_S$ , que representan al subconjunto de vehículos primarios y secundarios respectivamente, de tal manera que  $K = K_P \cup K_S$ ; todos los vehículos tienen una capacidad homogénea  $C$ . Para modelar tiempos, se tienen un tiempo de espera  $t_{espera}$  en cada vértice  $i \in V$  establecido en 5 minutos; para cada destino la hora de llegada  $H_{llegada}$  se definió con respecto al horario de entrada de los alumnos en el proceso de clustering y además se estableció una hora mínima de salida  $H_{salida}$ . Para este modelo, el conjunto  $E$  incluye las paradas de espacios académicos, lo que implica que solo se utilice el subconjunto  $S$  para representar las paradas de ubicaciones de alumnos a diferencia del modelo 5.3.2.

### Variables de decisión

Las variables de decisión del modelo consisten primeramente de la variable continua  $f_{ijkd}$  que expresa el flujo de alumnos que van del vértice  $i$  al  $j$  por el vehículo  $k$  y que van al destino  $d$ ; la variable binaria  $x_{ijk}$  toma el valor de 1 si la arista de  $i$  a  $j$  es utilizado por  $k$  o 0 en caso contrario; mientras que  $y_{ik}$  toma el valor de 1 si el vehículo  $k$  incide en el vértice  $i$  o 0 en caso contrario; la variable continua  $h_{ik}$  expresa el tiempo en el que un vehículo  $k$  debe de salir desde su respectiva terminal  $i$ ; la variable continua  $rt_i$  representa al residuo total de

alumnos no transportado en el vértice  $i$ , para cualquier vértice de suministro ( $i \in S \cup F$ ); la variable  $rp_{id}$  representa el residuo parcial de alumnos no transportados en el vértice  $i$  que se dirigen al destino  $d$  para cualquier vértice de suministro y la variable auxiliar continua  $u_{ik}$  indica el orden del vértice  $i$  por el vehículo  $k$ .

### Función objetivo

La función multi-objetivo de la Ecuación 5.28 busca obtener el flujo máximo hacia los destinos correspondientes considerando un costo mínimo (tiempo) y reduciendo el tiempo de salida de cada vehículo; las ponderaciones utilizadas para dar mayor o menor prioridad a cada uno de estos objetivos lo determinan los parámetros  $\alpha$ ,  $\beta$  y  $\gamma$  respectivamente.

$$\text{máx } \alpha \sum_{k \in K} \sum_{d \in D} \sum_{i \in V \setminus \{d\}} \sum_{j \in \{d\}} f_{ijkd} - \beta \sum_{k \in K} \sum_{i \in V} \sum_{j \in V, i \neq j} c_{ij} x_{ijk} + \gamma \sum_{k \in K} \sum_{i \in F} h_{ik} \quad (5.28)$$

### Restricciones de VRP

Las restricciones utilizadas del área de VRP para el caso de estudio se componen de la Ecuación 5.29 que distribuye el conjunto de vehículos principales a cada una de las terminales; la Ecuación 5.30 permiten distribuir el conjunto de vehículos secundarios a las terminales con base en el tamaño de dicho conjunto; las Ecuación 5.31 permite determinar la cantidad de vehículos que pueden visitar las paradas con respecto al volumen de alumnos; las Ecuaciones 5.32 y 5.33 determina el rango de destinos que puede visitar cada vehículo y el máximo de vehículos por destino respectivamente; la Ecuaciones 5.34, 5.35, 5.36, 5.37, 5.38 y 5.39 permiten modelar el comportamiento de los aristas entrantes para cada tipo de vértice, considerando que los vértices terminales no tienen aristas entrantes; las Ecuaciones 5.40, 5.41, 5.42 permite la eliminación de subciclos con base en la restricción MTZ (Miller-Tucker-Zemlin); la Ecuación 5.43 permite evitar loops (también conocidos como lazos o bucles en teoría de gráficas).

$$y_{ik} = 1 \quad \forall k \in K_P, i = F_k \quad (5.29)$$

$$\begin{cases} \sum_{i \in F} y_{ik} = 1 \quad \forall k \in K_S, & \text{si } |K_S| < |F| \\ 1 \leq \sum_{k \in K_S} y_{ik} \leq \left\lceil \frac{|K_S|}{|F|} \right\rceil \quad \forall i \in F, & \text{caso contrario} \end{cases} \quad (5.30)$$

$$\sum_{k \in K} y_{ik} = \left\lceil \frac{B_i}{C} \right\rceil \quad \forall i \in S \quad (5.31)$$

$$\sum_{i \in D} y_{ik} = 1 \quad \forall k \in K \quad (5.32)$$

$$1 \leq \sum_{k \in K} y_{ik} \leq \left\lceil \frac{|K|}{|D|} \right\rceil \quad \forall i \in D \quad (5.33)$$

$$\sum_{j \in V, j \neq i} x_{jik} = 0 \quad \forall i \in F, \forall k \in K \quad (5.34)$$

$$\sum_{j \in V \setminus F, j \neq i} x_{ijk} = y_{ik} \quad \forall i \in F, \forall k \in K \quad (5.35)$$

$$\sum_{j \in V, j \neq i} x_{jik} = y_{ik} \quad \forall i \in S, \forall k \in K \quad (5.36)$$

$$\sum_{j \in V \setminus F, j \neq i} x_{ijk} = y_{ik} \quad \forall i \in S, \forall k \in K \quad (5.37)$$

$$\sum_{j \in V, j \neq i} x_{jik} = y_{ik} \quad \forall i \in D, \forall k \in K \quad (5.38)$$

$$\sum_{j \in V, j \neq i} x_{ijk} = 0 \quad \forall i \in D, \forall k \in K \quad (5.39)$$

$$u_{ik} = 1, \quad \forall i \in F, \forall k \in K \quad (5.40)$$

$$u_{ik} - u_{jk} + nx_{ijk} \leq n - 1 \quad \forall i, j \in V \setminus F, i \neq j, \forall k \in K \quad (5.41)$$

$$1 \leq u_{ik} \leq n - 1 \quad \forall i, j \in V \setminus F, i \neq j, \forall k \in K \quad (5.42)$$

$$x_{ijk} = 0 \quad \forall i, j \in V, i = j, \forall k \in K \quad (5.43)$$

## Restricciones de Flujo Máximo

Para utilizar flujo máximo y permitir al modelo que cada vehículo lleve la máxima cantidad de alumnos hacia su destino correspondiente, la Ecuación 5.44 define la relación entre la variable de decisión  $x_{ijk}$  y  $f_{ijkd}$ , permitiendo establecer que el flujo máximo por la arista en la gráfica sea el valor de  $C$ ; de igual manera la Ecuación 5.45 establece la máxima capacidad para cada vehículo al trasladarse desde cualquier vértice; la Ecuación 5.46 determina que no existe flujo entrante a las terminales; la Ecuación 5.47 establece el comportamiento del flujo total saliente con respecto al entrante, para los vértices con suministro, considerando un residuo total de alumnos que no pueden ser trasladados; la Ecuación 5.48 establece el flujo parcial saliente y entrante para cada tipo de destino considerando un residuo parcial en cada vértice de suministro; la Ecuación 5.49 permite calcular el flujo total entrante hacia el destino indicado, considerando el residuo de alumnos que no se trasladaron por dicho flujo; la Ecuación 5.50 establece que el flujo de entrada de alumnos por cada vehículo en los vértices de suministro es menor o igual al flujo de salida para cada uno de los vehículos; con la Ecuación 5.51 se modela el comportamiento del flujo de salida de alumnos para cada vehículo que llegaron al destino que le corresponde; mientras que la Ecuación 5.52 establece el comportamiento del flujo total de salida con respecto al flujo total de entrada para cada destino, siendo el flujo de salida menor al ser vértices de demanda; la Ecuación

5.53 determina que no hay flujo de un vértice hacia si mismo; las Ecuaciones 5.54, 5.55, 5.56, 5.57, 5.58 y 5.59 establecen el rango de las variables de decisión.

$$\sum_{d \in D} f_{ijkd} \leq Cx_{ijk} \quad \forall i, j \in V, i \neq j, \forall k \in K \quad (5.44)$$

$$\sum_{j \in V \setminus F, j \neq i} \sum_{d \in D} f_{ijkd} \leq C \quad \forall i \in V, \forall k \in K \quad (5.45)$$

$$\sum_{j \in V} f_{jikd} = 0 \quad \forall i \in F, \forall k \in K, \forall d \in D \quad (5.46)$$

$$\sum_{k \in K} \sum_{d \in D} \sum_{j \in V \setminus F, j \neq i} f_{ijkd} - \sum_{k \in K} \sum_{d \in D} \sum_{j \in V, j \neq i} f_{jikd} + rt_i = B_i \quad \forall i \in S \cup F \quad (5.47)$$

$$\sum_{k \in K} \sum_{j \in V \setminus F} f_{ijkd} - \sum_{k \in K} \sum_{j \in V} f_{jikd} + rp_{id} = b_{id} \quad \forall i \in S \cup F, \forall d \in D \quad (5.48)$$

$$\sum_{k \in K} \sum_{j \in V, j \neq i} f_{jikd} + \sum_{i \in S \cup F} rp_{id} = B_d \quad \forall i, d \in D, i = d \quad (5.49)$$

$$\sum_{d \in D} \sum_{j \in V} f_{jikd} \leq \sum_{d \in D} \sum_{j \in V \setminus F} f_{ijkd} \quad \forall i \in S \cup F, \forall k \in K \quad (5.50)$$

$$\sum_{j \in V \setminus F, j \neq i} f_{ijkd} = 0 \quad \forall i, d \in D, i = d, \forall k \in K \quad (5.51)$$

$$\sum_{k \in K} \sum_{d \in D} \sum_{j \in V \setminus F, j \neq i} f_{ijkd} \leq \sum_{k \in K} \sum_{d \in D} \sum_{j \in V, j \neq i} f_{jikd} \quad \forall i \in D \quad (5.52)$$

$$\sum_{d \in D} f_{ijkd} = 0 \quad \forall i, j \in V, i = j, \forall k \in K \quad (5.53)$$

$$x_{ijk} \in \{0, 1\} \quad \forall i, j \in V, \forall k \in K \quad (5.54)$$

$$y_{ik} \in \{0, 1\} \quad \forall i \in V, \forall k \in K \quad (5.55)$$

$$f_{ijkd} \in \mathbb{Z}^+ \quad 0 \leq f_{ijkd} \leq C, \quad \forall i, j \in V, \forall k \in K, \forall d \in D \quad (5.56)$$

$$rt_i \in \mathbb{Z}^+ \quad rt_i \geq 0, \quad \forall i \in V \setminus D \quad (5.57)$$

$$rp_{id} \in \mathbb{Z}^+ \quad rp_{id} \geq 0, \quad \forall i \in V \setminus D, \forall d \in D \quad (5.58)$$

$$u_{ik} \in \mathbb{Z}^+ \quad u_{ik} \geq 0, \quad \forall i \in V, \forall k \in K \quad (5.59)$$

### Restricción de tiempo de salida

El evitar que las rutas sean muy largas, ya que las restricciones de flujo máximo permiten que los vehículos pasen por una gran cantidad de vértices para recolectar la mayor cantidad de alumnos, se aplica mediante la restricción de la Ecuación 5.60 para garantizar que los vehículos lleguen a los destinos correspondientes en el horario de entrada de los alumnos a la vez que se considera el tiempo tomado por la ruta y un tiempo de espera en cada parada; mientras que en la Ecuación 5.61 se especifica el rango de la variable de la hora de salida, permitiendo definir un tiempo constante para la hora de salida mínima de los vehículos en sus terminales correspondientes.

$$h_{ik} + \sum_{i,j \in V, i \neq j} (c_{ij} + t_{espera})x_{ijk} \leq H_{llegada} \quad \forall k \in K \quad (5.60)$$

$$h_{ik} \in \mathbb{R}^+ \quad H_{salida} \leq h_{ik} \leq H_{llegada} \quad \forall i \in F, \forall k \in K \quad (5.61)$$

### Adaptación del modelo para el sentido de REGRESO

Con lo mostrado en el modelo de flujo máximo para las rutas de ida en la sección 5.3.3, un modelo similar para las rutas de regreso conserva el planteamiento de los conjuntos y de algunas restricciones, sin embargo algunas conllevan a un planteamiento diferente. Para cada vértice  $d \in D$  se tiene un suministro de alumnos denotado como  $B_d$  y para cada vértice  $i \in S \cup F$  existe una demanda de alumnos que vienen desde cada destino  $d \in D$ , expresado como  $b_{id}$ .

Variables de decisión. Al igual que el modelo de ida, las variables de decisión del modelo consisten de la variable continua  $f_{ijkd}$  que expresa el flujo de alumnos que van del vértice  $i$  al  $j$  por el vehículo  $k$  desde el destino  $d$ ; la variable binaria  $x_{ijk}$  toma el valor de 1 si la arista de  $i$  a  $j$  es utilizado por  $k$  o 0 en caso contrario; mientras que  $y_{ik}$  toma el valor de 1 si el vehículo  $k$  incide en el vértice  $i$  o 0 en caso contrario; la variable continua  $rt_d$  representa al residuo total de alumnos no transportado en el destino  $d \in D$ ; la variable  $rp_{id}$  representa el residuo parcial de alumnos faltantes en el vértice  $i$  que vienen desde cualquier destino  $d \in D$  y la variable auxiliar continua  $u_{ik}$  indica el orden del vértice  $i$  por el vehículo  $k$ .

Función objetivo. La función multi-objetivo utiliza los mismos parámetros que en la Ecuación 5.28, pero su formulación difiere, ya que ahora se busca el flujo máximo en todas las aristas de la gráfica a un costo mínimo, ver Ecuación 5.62.

$$\text{máx } \alpha \sum_{d \in D} \sum_{k \in K} \sum_{i \in V} \sum_{j \in V, i \neq j} f_{ijkd} - \beta \sum_{k \in K} \sum_{i \in V} \sum_{j \in V, i \neq j} c_{ij}x_{ijk} + \gamma \sum_{k \in K} \sum_{i \in F} h_{ik} \quad (5.62)$$

Restricciones de VRP. Se conservan las restricciones de incidencia de vehículos a vértices, planteadas en las Ecuaciones 5.29, 5.30, 5.31, 5.32 y 5.33; las Ecuaciones 5.34, 5.35, 5.36, 5.37, 5.38 se modifican para poder representar el comportamiento de las aristas de entrada y salida para el sentido de regreso, considerando que los vértices destinos no tienen aristas

entrantes y los vértices terminales no tienen aristas salientes, por lo que las Ecuaciones 5.63, 5.64, 5.65, 5.66, 5.67 y 5.68 expresan dichos cambios; las Ecuaciones 5.69, 5.70, 5.71 permite la eliminación de subciclos para el sentido de regreso.

$$\sum_{j \in V, j \neq i} x_{jik} = 0 \quad \forall i \in F, \forall k \in K \quad (5.63)$$

$$\sum_{j \in V \setminus F, j \neq i} x_{ijk} = y_{ik} \quad \forall i \in F, \forall k \in K \quad (5.64)$$

$$\sum_{j \in V, j \neq i} x_{jik} = y_{ik} \quad \forall i \in S, \forall k \in K \quad (5.65)$$

$$\sum_{j \in V \setminus F, j \neq i} x_{ijk} = y_{ik} \quad \forall i \in S, \forall k \in K \quad (5.66)$$

$$\sum_{j \in V, j \neq i} x_{jik} = y_{ik} \quad \forall i \in D, \forall k \in K \quad (5.67)$$

$$\sum_{j \in V, j \neq i} x_{ijk} = 0 \quad \forall i \in D, \forall k \in K \quad (5.68)$$

$$u_{ik} = 1, \quad \forall i \in D, \forall k \in K \quad (5.69)$$

$$u_{ik} - u_{jk} + nx_{ijk} \leq n - 1 \quad \forall i, j \in V \setminus D, i \neq j, \forall k \in K \quad (5.70)$$

$$1 \leq u_{ik} \leq n - 1 \quad \forall i, j \in V \setminus D, i \neq j, \forall k \in K \quad (5.71)$$

Restricciones de Flujo Máximo. Para utilizar flujo máximo en el modelo de regreso, se aplican de igual manera la restricción de la Ecuación 5.44, sin embargo todas las ecuaciones posteriores se modifican, por lo que la Ecuación 5.72 establece la máxima capacidad para cada vehículo al trasladarse desde cualquier vértice; la Ecuación 5.73 determina que no existe flujo entrante a los destinos; la Ecuación 5.74 establece el comportamiento del flujo total saliente con respecto al entrante, para los vértices con demanda, considerando un residuo total de alumnos que no pueden ser llevados; la Ecuación 5.75 establece el flujo parcial saliente y entrante para cada tipo de destino, considerando un residuo parcial en cada vértice de demanda; la Ecuación 5.76 permite calcular el flujo total entrante hacia el destino indicado, considerando el residuo de alumnos que no se trasladaron por dicho flujo; la Ecuación 5.77 establece que el flujo de entrada de alumnos por cada vehículo en los vértices de suministro es menor o igual al flujo de salida por cada uno de los vehículos; con la Ecuación 5.78 se modela el comportamiento del flujo de salida de alumnos para cada vehículo que llegaron al destino que le corresponde; mientras que la Ecuación 5.79 establece el comportamiento del flujo total de salida con respecto al flujo total de entrada para cada destino, siendo el flujo de salida menor al ser vértices de demanda.

$$\sum_{j \in V \setminus D, j \neq i} \sum_{d \in D} f_{ijkd} \leq C \quad \forall i \in V, \forall k \in K \quad (5.72)$$

$$\sum_{j \in V} f_{jikd} = 0 \quad \forall i \in D, \forall k \in K, \forall d \in D \quad (5.73)$$

$$\sum_{k \in K} \sum_{d \in D} \sum_{j \in V \setminus F, j \neq i} f_{jikd} - \sum_{k \in K} \sum_{d \in D} \sum_{j \in V, j \neq i} f_{jikd} + rt_i = B_i \quad \forall i \in S \cup F \quad (5.74)$$

$$\sum_{k \in K} \sum_{j \in V \setminus F} f_{jikd} - \sum_{k \in K} \sum_{j \in V} f_{jikd} + rp_{id} = b_{id} \quad \forall i \in S \cup F, \forall d \in D \quad (5.75)$$

$$\sum_{k \in K} \sum_{j \in V, j \neq i} f_{jikd} + \sum_{i \in S \cup F} rp_{id} = B_d \quad \forall i, d \in D, i = d \quad (5.76)$$

$$\sum_{d \in D} \sum_{j \in V} f_{jikd} \leq \sum_{d \in D} \sum_{j \in V \setminus F} f_{jikd} \quad \forall i \in S \cup F, \forall k \in K \quad (5.77)$$

$$\sum_{j \in V \setminus F, j \neq i} f_{jikd} = 0 \quad \forall i, d \in D, i = d, \forall k \in K \quad (5.78)$$

$$\sum_{k \in K} \sum_{d \in D} \sum_{j \in V \setminus F, j \neq i} f_{jikd} \leq \sum_{k \in K} \sum_{d \in D} \sum_{j \in V, j \neq i} f_{jikd} \quad \forall i \in D \quad (5.79)$$

## 5.4. Visualización de las rutas obtenidas por los modelos

Los resultados obtenidos por los modelos de ILP propuestos se muestran en las siguientes secciones, en el modelo de dos índices se muestran las rutas generadas y los resultados obtenidos en cuanto a volumen y distancias recorridas; para los modelos que implican flujo son descritos los escenarios de pruebas, así como los parámetros establecidos para su ejecución.

### 5.4.1. Resultados del modelo de CVRP a dos índices

Del modelo a dos índices descrito en 5.3.2 y utilizando los datos obtenidos de la salida de los procesos de clustering mostrados en la sección 5.2, se obtuvieron las rutas mostradas en la Figura 5.13; dónde las rutas trazadas no terminan en algún depósito, como los modelos clásicos de VRP o sus variaciones mostrados en la sección 2.4, los puntos de término pueden ser las distintas *estaciones* definidas en el subconjunto  $E$  o el depósito mismo, denotado como el vértice  $\{0\}$ .

En la Tabla 5.7 se puede observar las rutas, la distancia que abarcan en kilómetros y la cantidad de alumnos que pueden transportarse en cada una. El valor de la función objetivo alcanzado con este modelo fue de 3.3062 que equivale a una distancia de 330.62 km; el tiempo de ejecución tomado para resolver este modelo de ILP fue de en promedio de 00 : 03 : 04 min, con lo cual es completamente viable utilizar este modelo a dos índices para la determinación de las rutas óptimas de transporte en un sistema web, sin embargo el modelo aún necesita restricciones para adecuarse más a la realidad y determinar el número de camiones necesarios.

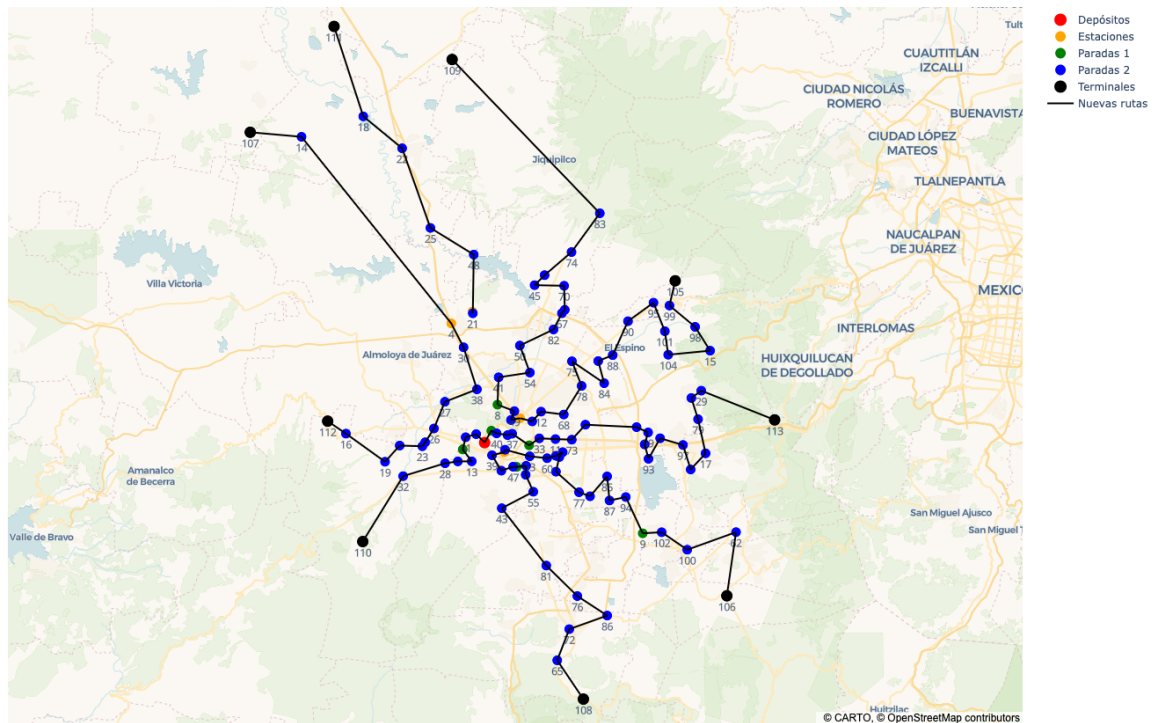


Figura 5.13: Rutas obtenidas con un modelo de ILP de dos índices (Elaboración propia).

Tabla 5.7: Valoración de las rutas obtenidas del modelo a dos índices (Elaboración propia).

No. Ruta	Ruta	Distancia (km)	Volumen de alumnos
1	107, 100, 99, 14, 101, 106, 105, 93, 96, 91, 87, 78, 81, 61, 58, 21, 2	43.12	1210
2	108, 64, 85, 103, 9, 92, 90, 88, 83, 80, 67, 69, 71, 66, 63, 53, 5, 37, 45, 44, 39, 7, 0	43.71	910
3	109, 13, 4	28.8	84
4	110, 68, 75, 89, 79, 84, 42, 56, 51, 52, 6, 47, 41, 38, 43, 1	39.24	859
5	111, 86, 77, 60, 35, 73, 74, 70, 11, 50, 54, 40, 8, 49, 46, 2	54.41	1062
6	112, 22, 27, 29, 31, 10, 30, 48, 0	19.68	600
7	113, 17, 20, 24, 33, 23, 3	33.22	227
8	114, 15, 18, 19, 12, 32, 25, 26, 34, 59, 4	26.73	961
9	115, 28, 36, 72, 16, 62, 102, 104, 97, 95, 98, 94, 82, 76, 65, 57, 55, 2	41.7	790

Como un primer análisis entre éste modelo y las rutas de transporte actuales, en la Figura 5.14 se puede observar las rutas actualmente operativas graficadas sobre un plano que contienen los puntos de parada identificados por los procesos de clustering y el Convex Hull.

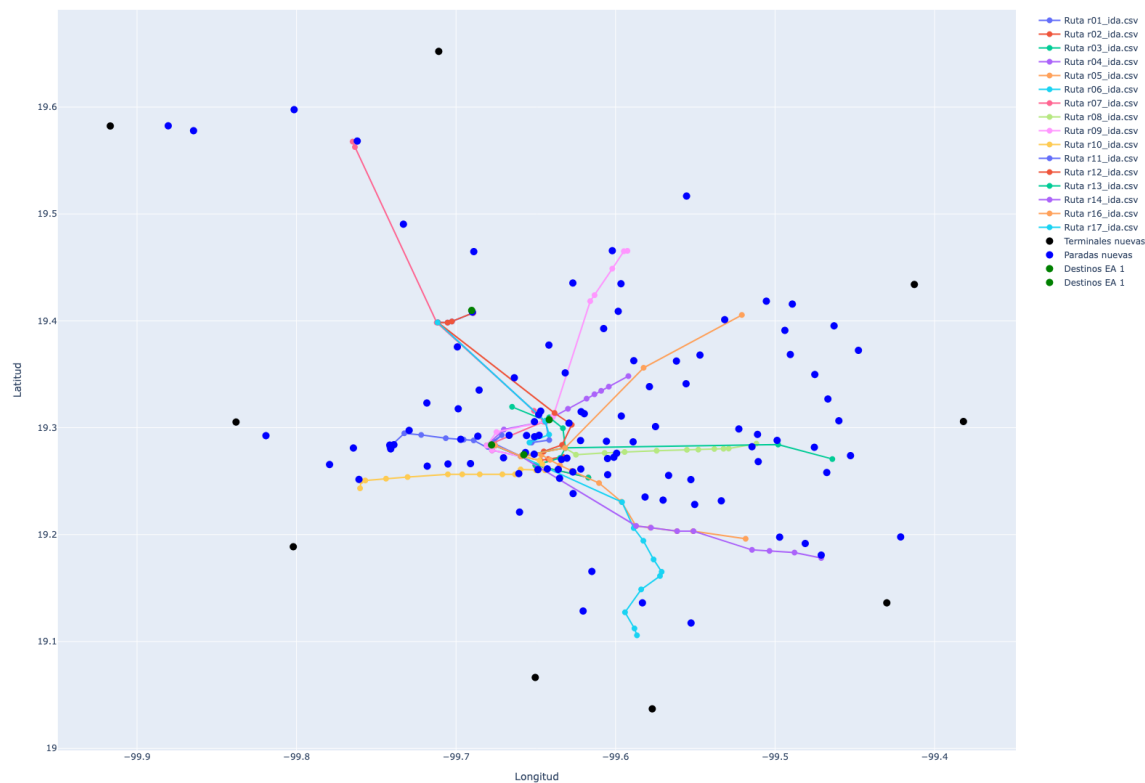


Figura 5.14: Rutas actuales y puntos de parada del proceso de clustering (Elaboración propia).

Comparando las rutas de las Figuras 5.13 y 5.14 podemos notar que hay cierta convergencia entre la mayoría de rutas actuales y los nuevos puntos de parada. Sin embargo, notamos algunas diferencias, principalmente que los puntos terminales y algunas paradas tienden a estar más alejadas que los inicios de las rutas que se encuentran operativas; además hay algunos puntos de parada que no pertenecen a algunos de las rutas actuales. Esta situación no debe considerarse como una problemática o inconveniente por parte de las tres etapas de clustering, ya que demuestra que dado el volumen de alumnos, es conveniente tener inicios de ruta más alejados de los destinos a donde los alumnos se dirigen. De igual manera, también puede hacerse una toma de decisiones para definir si esos puntos serán los definitivos o serán cambiados antes de su uso por parte de alguno de los modelos de ILP, propuestos en las secciones 5.3.2 y 5.3.3, para finalmente arrojar un nuevo conjunto de

rutas de transporte.

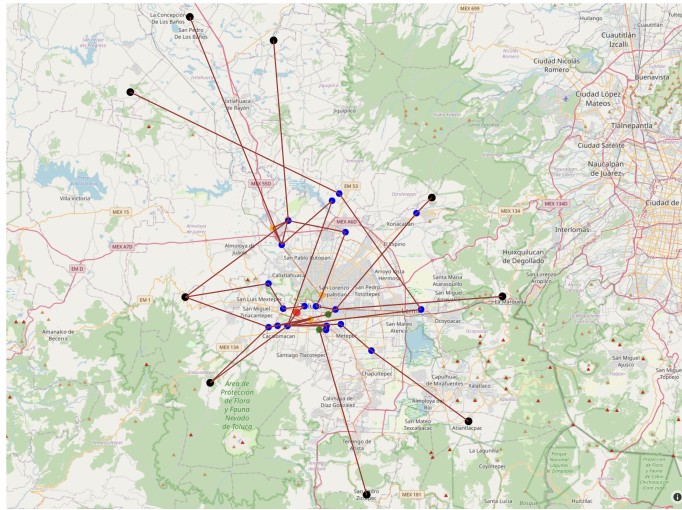
#### 5.4.2. Resultados del modelo de IDA a flujo máximo

Del modelo de ILP propuesto para el caso de las rutas de IDA en la sección 5.3.3 y utilizando los datos obtenidos de la salida de los procesos de clustering descritos en 5.2, se obtuvieron las rutas mostradas en la imagen de la Figura 5.15; dónde cada una de las rutas no terminan en algún depósito (campus universitario), como los modelos clásicos de VRP o sus variaciones, ya que los puntos de término pueden ser los distintos *destinos* definidos en el subconjunto  $D$ .

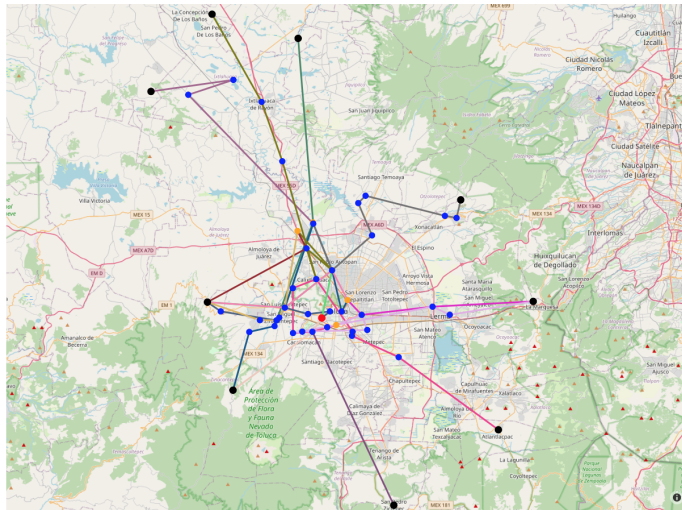
Las pruebas al modelo se realizaron con tres escenarios tomados del mismo conjunto de datos determinados por los procesos de clustering, la resolución del modelo para cada uno de los escenarios se realizó con el solver COIN-OR Branch and Cut (CBC) de Programación Lineal Mixta versión 2.10.8, mediante la ejecución de éste sobre un Servidor Virtual Privado (VPS, por sus siglas en inglés) con 8 GB de memoria RAM y un procesador de 32 núcleos AMD EPYC 9354P a 2.0 GHz. Mediante la Tabla 5.8 se detallan las dimensiones y configuraciones de cada escenario.

Tabla 5.8: Configuraciones de los escenarios de pruebas (Elaboración propia).

Escenario	Número de Vehículos	Cantidad de vértices			Tiempo máximo para el solver
		Paradas	Destinos	Terminales	
1	15	20	3	9	1,200 seg
2	15	35	5	9	604,800 seg
3	45	104	11	9	2,592,000 seg



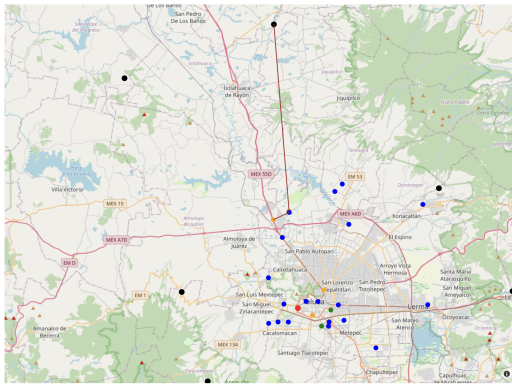
(a) Rutas de IDA obtenidas con el modelo de ILP para el escenario 1.



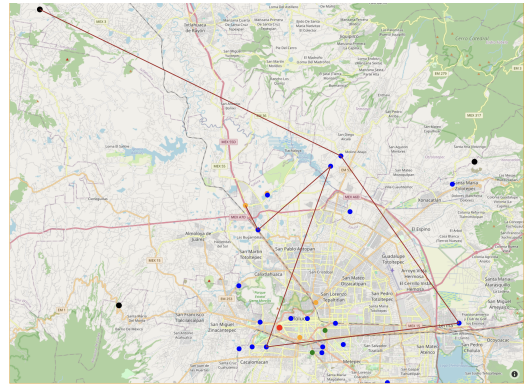
(b) Rutas de IDA obtenidas con el modelo de ILP para el escenario 2.

Figura 5.15: Rutas de IDA obtenidas en un horario de 7:00 am para los escenarios 1 y 2 (Elaboración propia).

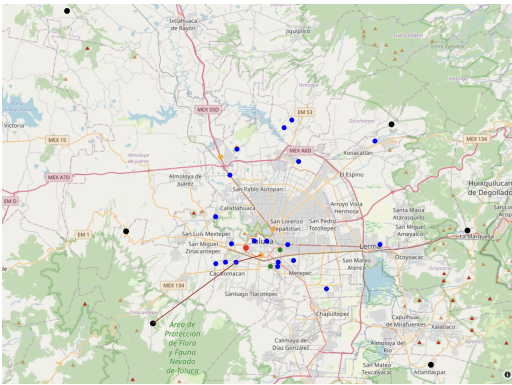
Una muestra de las rutas individuales por escenario se pueden apreciar en las Figuras 5.16 y 5.17, observando diferentes trazados al variar la cantidad de vehículos, vértices y destinos.



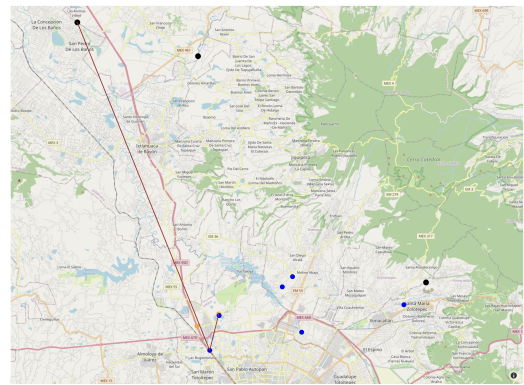
(a) Ruta del vehículo 1.



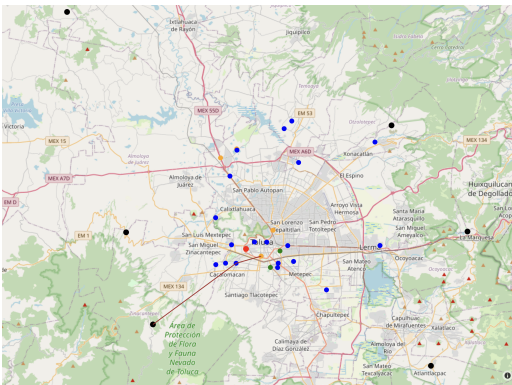
(b) Ruta del vehículo 2.



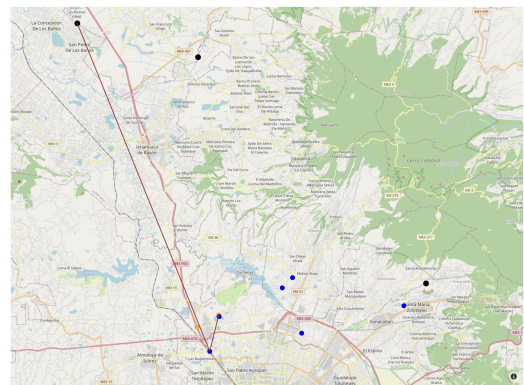
(c) Ruta del vehículo 3.



(d) Ruta del vehículo 4.

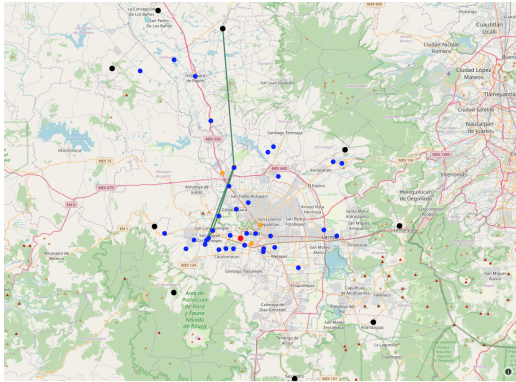


(e) Ruta del vehículo 5.

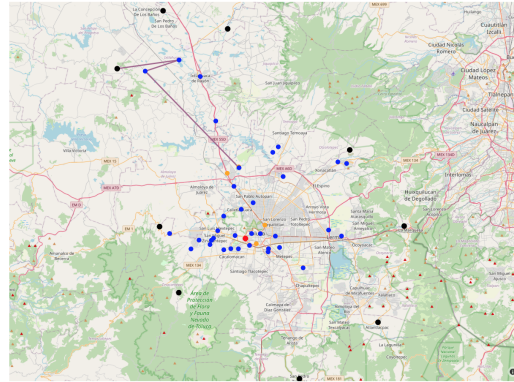


(f) Ruta del vehículo 6.

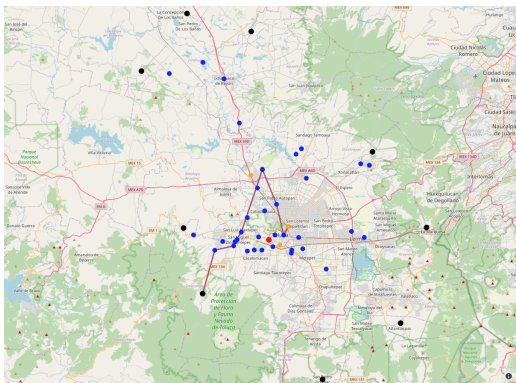
Figura 5.16: Muestra de las rutas de IDA obtenidas en un horario de 7:00 am para el escenario 1 (Elaboración propia).



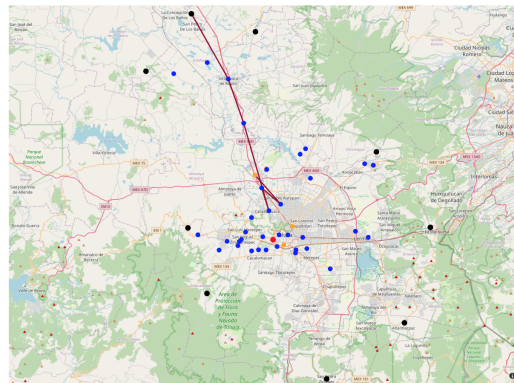
(a) Ruta del vehículo 1.



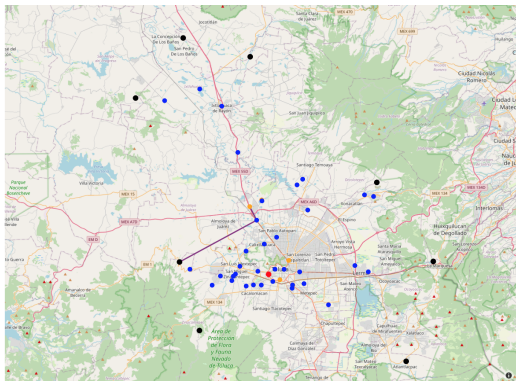
(b) Ruta del vehículo 2.



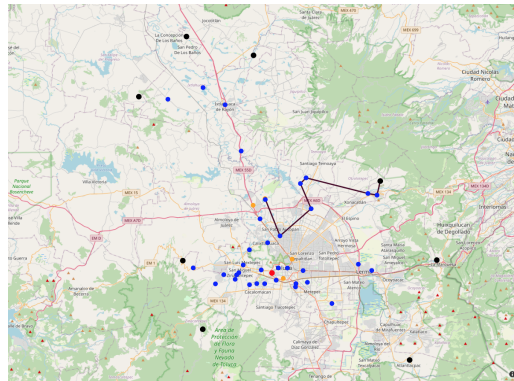
(c) Ruta del vehículo 3.



(d) Ruta del vehículo 4.



(e) Ruta del vehículo 5.



(f) Ruta del vehículo 6.

Figura 5.17: Muestra de las rutas de IDA obtenidas en un horario de 7:00 am para el escenario 2 (Elaboración propia).

El valor de la función objetivo alcanzado para cada escenario con el modelo de flujo máximo, la cantidad de nodos enumerados, brecha de optimalidad (GAP), número de iteraciones y el tiempo en segundos utilizado por el solver CBC para determinar la primera solución entera encontrada, se muestran en la Tabla 5.9. Para los escenarios 1 y 2, fue posible obtener una solución entera por parte del solver CBC, mientras que para el escenario 3, el tiempo propuesto no fue suficiente.

Tabla 5.9: Resultados de las pruebas con los distintos escenarios (Elaboración propia).

<b>Escenario</b>	<b>Función Objetivo</b>	<b>Nodos Enumerados</b>	<b>GAP</b>	<b>Iteraciones</b>	<b>Tiempo CPU</b>
1	1,040.632	109	-0.12	160,803	241.57 seg
2	514.2122	324	-0.04	3,869,554	14,895 seg
3	-	-	-	-	-

Para cada escenario, en la Tabla 5.10, se muestran las distancias totales de las rutas obtenidas, así como el volumen de alumnos transportados hacia cada uno de los destinos establecidos, al tener diferentes cantidades de vehículos y puntos en cada escenario los volúmenes de alumnos muestran valores diversos.

Tabla 5.10: Resultados de las pruebas con los distintos escenarios (Elaboración propia).

<b>Escenario</b>	<b>Tiempo Total</b>	<b>Distancia Total</b>	<b>Flujo de alumnos por destino</b>				
			$D_1$	$D_2$	$D_3$	$D_4$	$D_5$
1	15:53	505.72 km	199	178	113	-	-
2	29:02	894.67 km	180	120	180	180	151
3	-	-	-	-	-	-	-

En las Tablas 5.11 y 5.12 se muestran las rutas, distancias que abarcan en kilómetros y la cantidad de alumnos que se transportaron hacia sus destinos para los escenarios 1 y 2 por cada ruta determinada por el modelo.

Tabla 5.11: Rutas obtenidas del escenario 1 y su valoración (Elaboración propia).

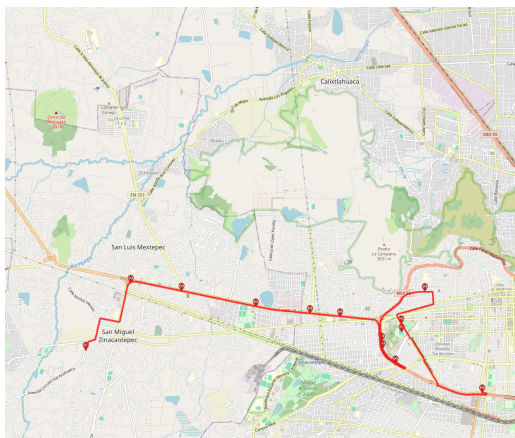
<b>Vehículo</b>	<b>Ruta</b>	<b>Distancia (km)</b>	<b>Tiempo trayecto (hrs)</b>	<b>Hora Salida</b>	<b>Volumen de alumnos</b>
1	23,9,5,0	41.42	01:17	5:43 am	60
2	24,12,14,5,0	61.11	01:52	5:08 am	60
3	25,0	16.8	00:30	6:30 am	2
4	26,9,8,2	50.23	01:30	5:30 am	19
5	27,4,11,16,18,1	30.02	01:10	5:50 am	60
6	28,20,17,9,2	35.71	01:14	5:46 am	60
7	29,19,0	34.93	01:02	5:58 am	17
8	30,22,13,2	29.45	00:59	6:01 am	3
9	31,21,1	29.15	00:54	6:06 am	10
10	24,10,1	46.06	01:19	5:41 am	60
11	31,1	27.25	00:46	6:14 am	6
12	25,10,6,2	24.07	00:51	6:09 am	31
13	30,10,0	33.32	01:00	6:00 am	60
14	27,7,3,1	28.30	00:57	6:03 am	40
15	25,1	17.9	00:32	6:28 am	2

Tabla 5.12: Rutas obtenidas del escenario 2 y su valoración (Elaboración propia).

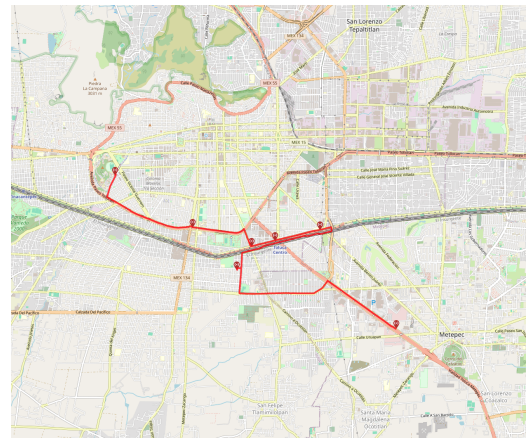
<b>Vehículo</b>	<b>Ruta</b>	<b>Distancia (km)</b>	<b>Tiempo trayecto (hrs)</b>	<b>Hora Salida</b>	<b>Volumen de alumnos</b>
1	40,25,15,9,3	79.71	02:20	5:13 am	60
2	41,7,5,25,3	106.08	02:59	5:32 am	60
3	42,6,10,9,15,25,21,22,2	130.5	03:56	5:14 am	60
4	43,11,14,22,18,21,4	54.28	01:51	5:01 am	60
5	44,18,4	20.05	00:40	6:21 am	31
6	45,39,35,28,33,30,21,3	69.2	02:19	5:27 am	60
7	46,26,23,1	18.53	00:43	5:57 am	60
8	47,37,31,32,22,16,20,26,1	71.64	02:27	5:01 am	60
9	48,38,34,2	21.24	00:47	6:03 am	60
10	44,8,9,15,17,24,0	101.22	03:02	5:58 am	60
11	42,26,0	7.65	00:21	6:25 am	60
12	44,15,21,0	18.66	00:43	6:04 am	60
13	44,15,21,2	23.27	00:50	6:07 am	60
14	48,36,38,20,19,29,34,1	76.09	02:29	5:05 am	60
15	44,12,9,15,4	96.55	02:45	6:04 am	60

## 5.5. Comparativa de las rutas obtenidas y las rutas actuales

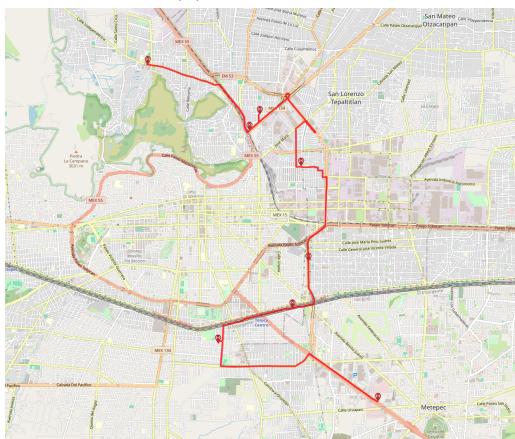
Con las rutas obtenidas por el modelo matemático, descrito en la sección 5.3.3, se puede realizar un análisis y comparativa con las actuales rutas de transporte. Primeramente podemos retomar lo descrito sobre el servicio actual, de la sección 1.1, de lo cual se sabe que las rutas de transporte fueron generadas con análisis empíricos, estadísticos, encuestas, cuestionarios y algunas técnicas heurísticas como el método “de barrido”, mencionado en la sección 2.7.2. Para analizar visualmente las rutas actuales, en la Figura 5.18 se muestran cuatro rutas en sentido de IDA, donde se observan sus puntos de parada y el trayecto siguiendo el sentido de calles.



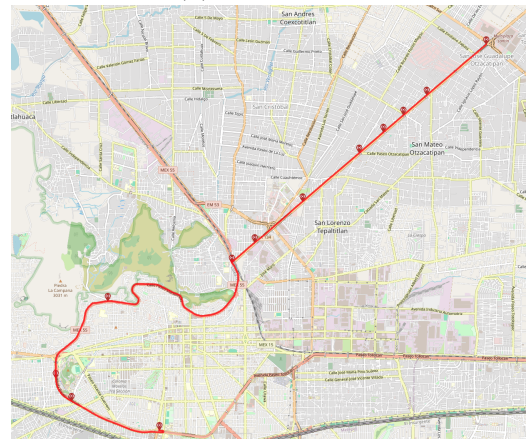
(a) Ruta No. 1.



(b) Ruta No. 2.



(c) Ruta No. 3.



(d) Ruta No. 4.

Figura 5.18: Muestra de las rutas de IDA actuales (Elaboración propia).

Con el modelo a flujo máximo solo fue posible obtener rutas por vehículo para los escenarios 1 y 2, en contraposición el escenario 3 que no pudo ser resuelto en un tiempo computacionalmente aceptable. Esto dificulta los elementos para realizar una comparativa adecuada entre las rutas actuales y las generadas por el modelo de ILP, mediante el uso de la función objetivo de dicho modelo, ver Ecuación 5.28,

Dada la saturación del servicio de transporte, por parte de los usuarios, las rutas actuales llevan un excedente de pasajeros en comparación a las rutas del modelo de flujo máximo, ya que las nuevas rutas evitan llevar más de un límite  $C$  de 60 pasajeros, además de garantizar que los usuarios recolectados lleguen al destino correcto. En cuanto al volumen de alumnos, las rutas actuales llevan más pasajeros en ciertos horarios, considerando que transportan más de 70 estudiantes por vehículo; sin embargo, dicha situación genera riesgos en el transporte mismo; dichas problemáticas son contempladas por el modelo mediante restricciones. Las rutas nuevas, al haber utilizado procesos de clustering y una envolvente convexa permiten tener puntos de inicio y paradas de ruta contemplando la ubicación de los alumnos; por otro lado, las paradas de las rutas actuales fueron posicionados mediante datos estadísticos, que genera ciertos problemas al no ser actualizados periódicamente debido a las variaciones en la distribución de los alumnos cada ciclo escolar.

Las distancias de las rutas nuevas son más extensas que las rutas actuales, esto se puede observar con las Tablas 5.12 y 5.13, lo que implica que recorren mayor trayecto y por consiguiente toman más tiempo; sin embargo, mediante restricciones de tiempo, se asegura de que lleguen a los destinos correspondientes de acuerdo a lo estipulado con los tiempos de entrada de los estudiantes.

Tabla 5.13: Distancias de las rutas actuales para el sentido de IDA (Elaboración propia).

Ruta	Distancia (km)	Número Paradas	Número de autobuses	Ruta	Distancia (km)	Número Paradas	Número de autobuses
1	15.8	13	3	9	29.6	11	2
2	13.1	7	2	10	23.2	17	1
3	18.0	8	3	11	19.0	8	8
4	17.0	12	4	12	22.2	8	6
5	21.0	12	1	13	27.3	5	2
6	27.2	12	3	14	27.7	10	2
7	41.4	8	1	15	27.0	5	1
8	23.2	13	5	16	16.6	4	1

## 5.6. Prototipo de sistema

Trasladar todos los elementos desarrollados, procesos clustering y modelos de ILP, a un sistema que permita generar rutas de transporte, necesitó de una selección de acciones que permita a los usuarios interactuar con dichos componentes, de manera sencilla y transparente. Con la Figura 5.19 se muestra un diagrama de casos de uso, que permite modelar las actividades principales que puede realizar un usuario para una propuesta de prototipo de sistema.

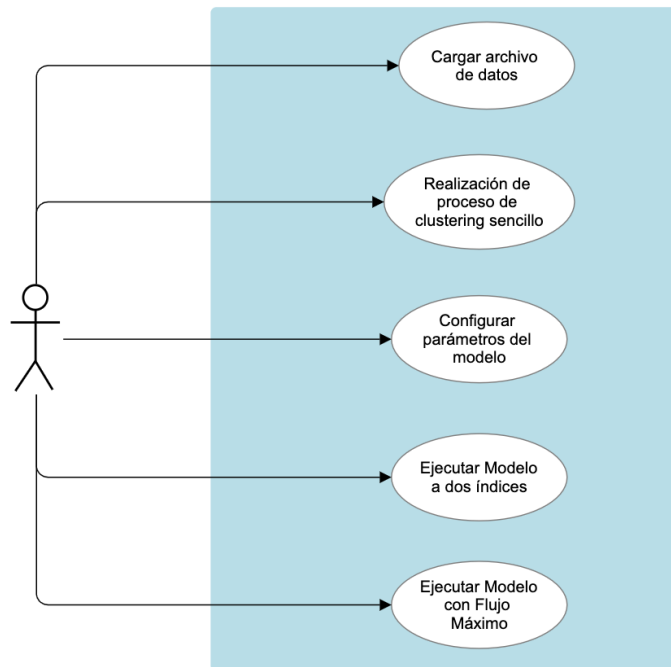


Figura 5.19: Diagrama de Casos de Uso para el prototipo de sistema (Elaboración propia).

Las descripciones de las acciones del usuario se presentan a continuación:

1. Cargar archivo de datos. Permite al usuario subir la matrícula de alumnos en formato CSV para ser procesada.
2. Procesos de clustering. El prototipo permite al usuario realizar el proceso de clustering, con los datos cargados, para configurar los parámetros básicos que utiliza DBSCAN, distancia Eps y MintPts.
3. Configuración de parámetros del modelo. El usuario tiene permitido configurar algunos parámetros del modelo: velocidad del vehículo, cantidad de vehículos, capacidad de

pasajeros por vehículo, asignar valores a los coeficientes  $\alpha$ ,  $\beta$  y  $\gamma$  para ponderar los componentes de la función objetivo.

4. Ejecutar Modelo a dos índices. Con los datos del proceso de clustering, se le permite al usuario ejecutar un modelo de CVRP de dos índices, con el cual se obtienen rutas óptimas y posteriormente ser mostradas en un mapa.
5. Ejecutar Modelo con Flujo Máximo. Con una porción de los datos del proceso de clustering, permite al usuario ejecutar un modelo de CVRP con flujo máximo, con el cual se obtienen rutas óptimas y posteriormente ser mostradas en un mapa.
6. Ver rutas actuales. El usuario puede ver las rutas que actualmente están en funcionamiento, tanto en sentido de IDA, REGRESO o ambos.

La propuesta para el desarrollo de este prototipo, se determinó para realizarse mediante un entorno web, bajo los lenguajes de programación javascript, PHP, Python 3.13.1 y Julia 1.11.4, los dos últimos fueron elegidos para implementar el proceso de clustering y los modelos de ILP respectivamente. La comunicación entre los proceso de preprocesamiento, clustering y los modelos de ILP se realizó mediante archivos en formato CSV y JSON.

En la Figura 5.20, se muestra la interfaz general del prototipo, destacando un menú lateral izquierdo para la ejecución de las funcionalidades principales: mostrar rutas actuales, proceso de clustering e iniciar el solver del modelo.

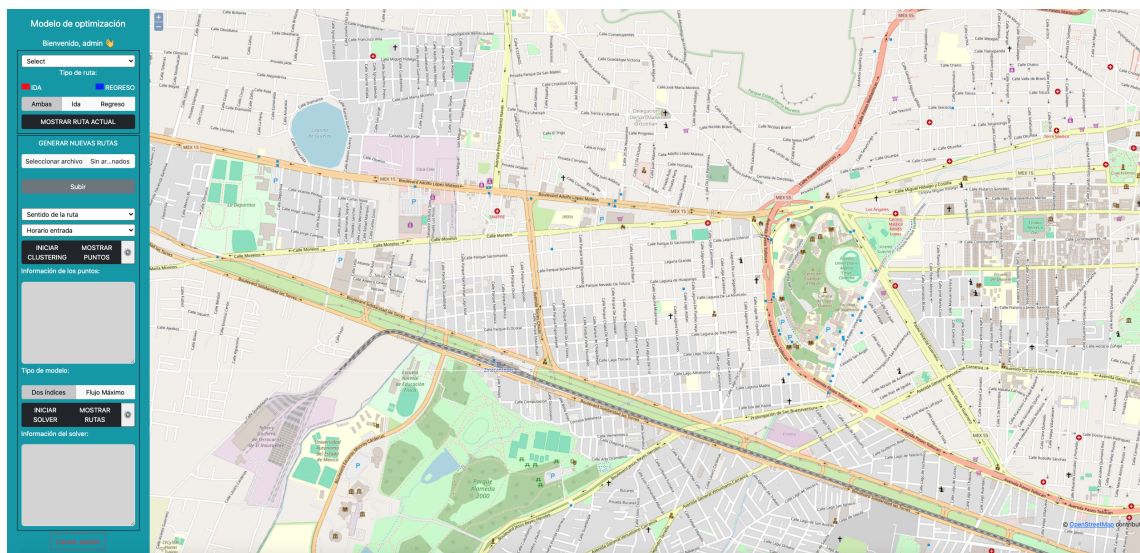
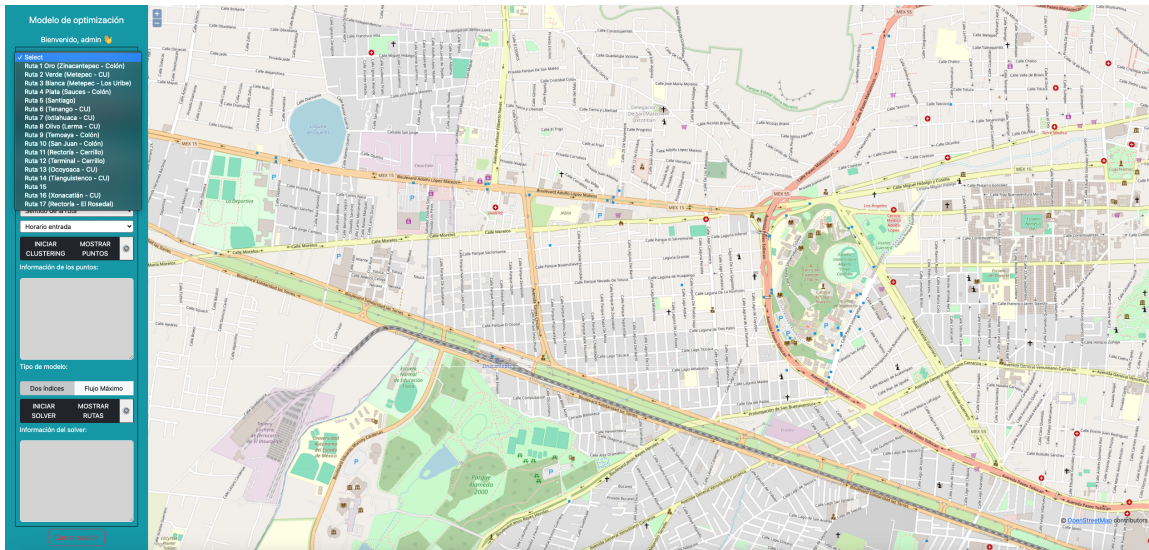
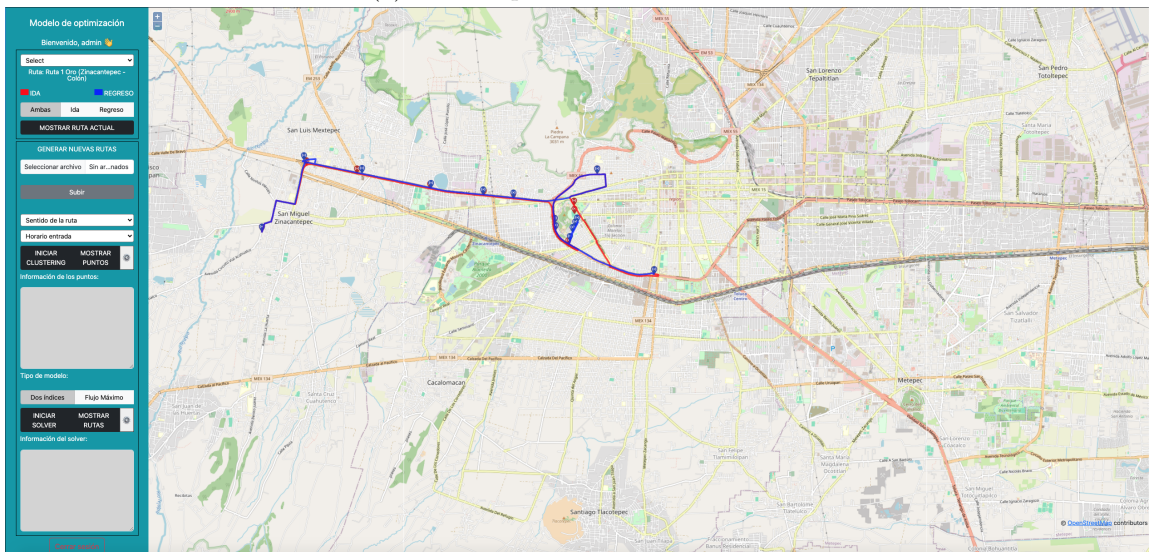


Figura 5.20: Interfaz general del prototipo de sistema (Elaboración propia).

En las Figuras 5.21a y 5.21b, se puede ver las funcionalidades para mostrar las 16 rutas actuales, tanto en los sentidos de IDA, REGRESO o ambos, mediante herramientas de trazado de rutas en mapas.



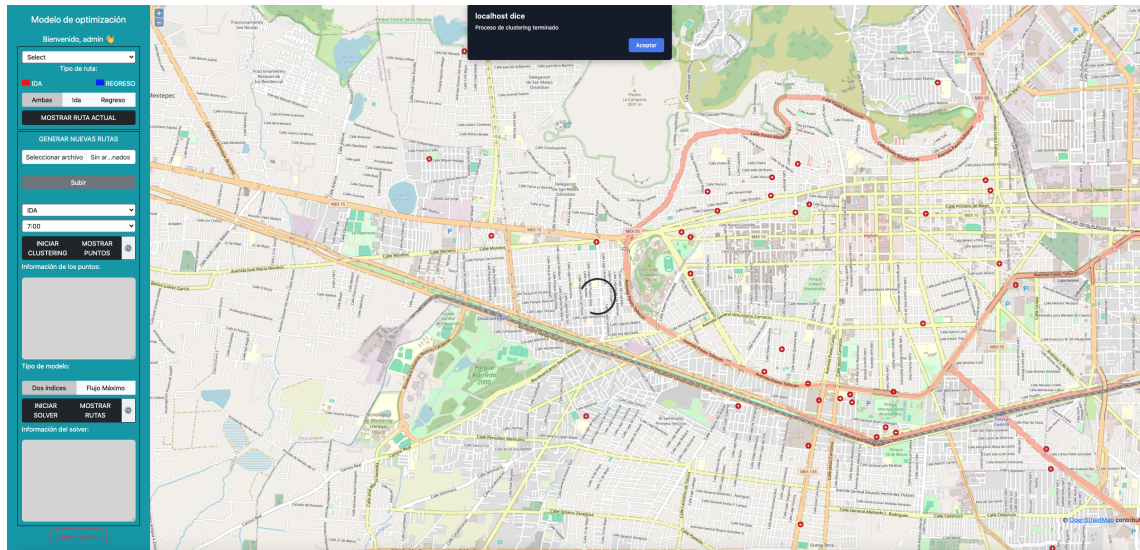
(a) Interfaz para las rutas actuales.



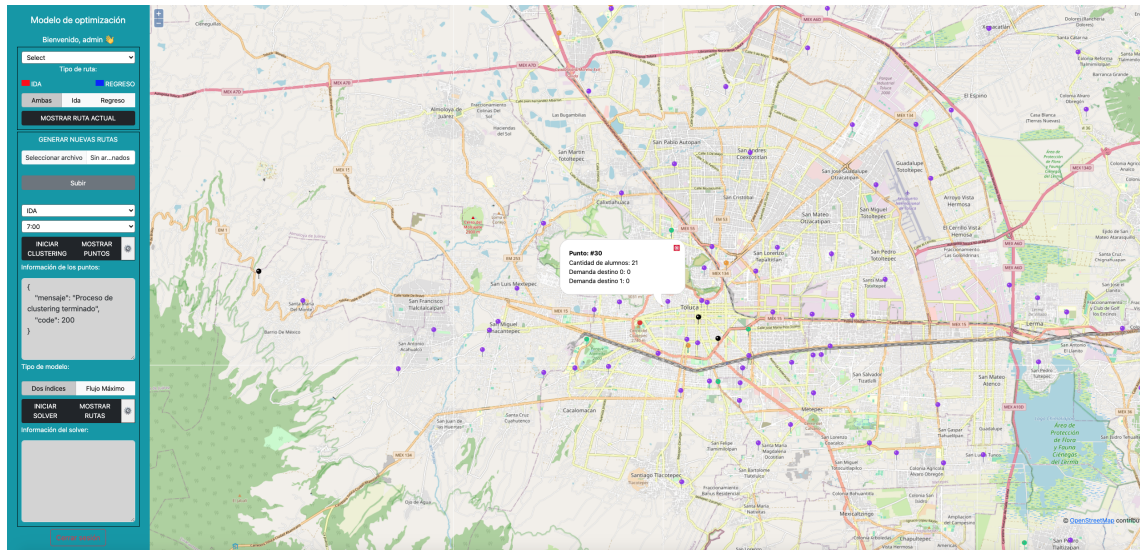
(b) Interfaz para las rutas actuales.

Figura 5.21: Interfaz para las rutas actuales (Elaboración propia).

Las Figuras 5.22a y 5.22b, muestran el proceso de clustering, en donde al obtener los puntos finales, estos se muestran en el mapa, permitiendo observar la información de cada punto como es: Id del punto, cantidad de alumnos y demandas por destino.



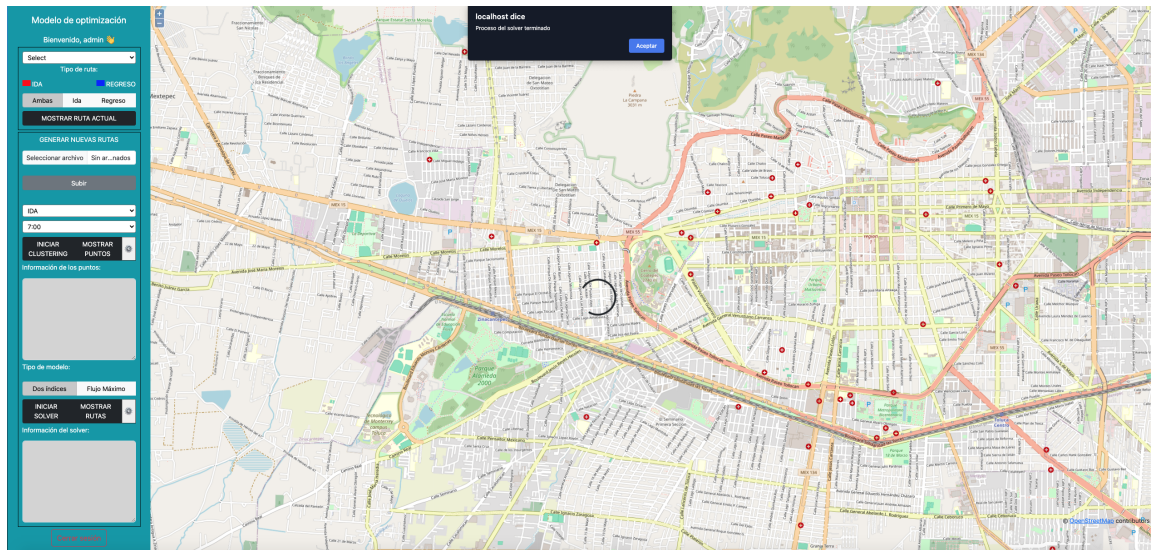
(a) Inicio del proceso de clustering.



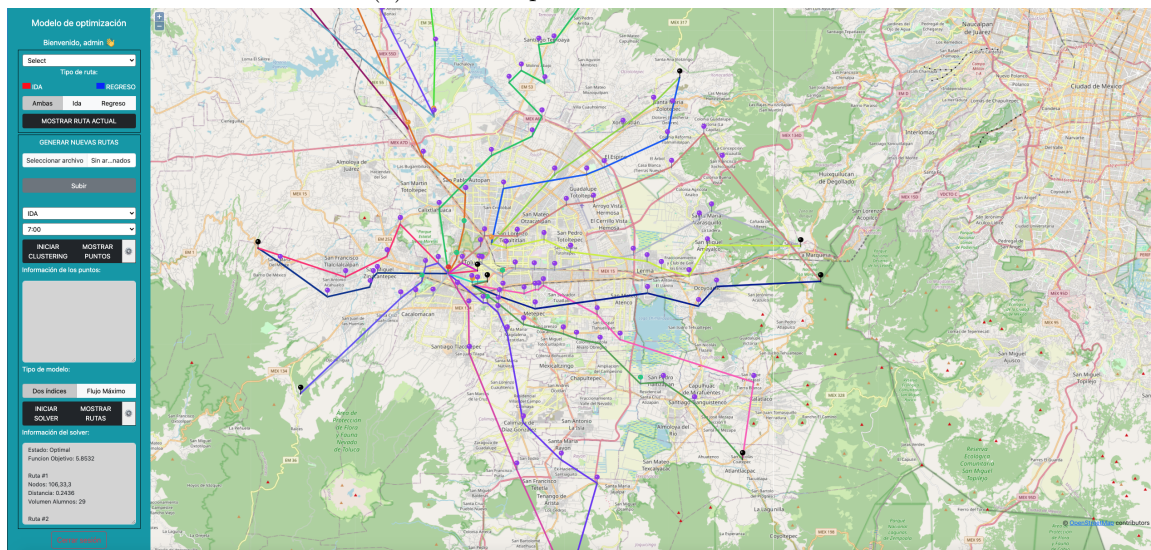
(b) Puntos encontrados.

Figura 5.22: Proceso de clustering (Elaboración propia).

El proceso de los solvers de ILP, de las secciones 5.3.2 y 5.3.3, trasladados el prototipo de sistema se muestran en las Figuras 5.23.



(a) Inicio del proceso de los solvers.



(b) Muestra de las nuevas rutas.

Figura 5.23: Prototipo de sistema (Elaboración propia).

Las configuraciones posibles para el proceso de clustering y para el modelo de flujo máximo, presentado en la sección 5.3.3, son mostradas mediante la Figura 5.24 y la Figura 5.25 respectivamente.

Para la configuración de los procesos de clustering, se permite establecer los parámetros de distancia (Eps) y puntos mínimos (MinPts) de DBSCAN, para cada una de las tres etapas, mediante un menú de configuración. Las distancias se manejan en km y para cada etapa se tienen rangos mínimos y máximos para facilitar dicha configuración.

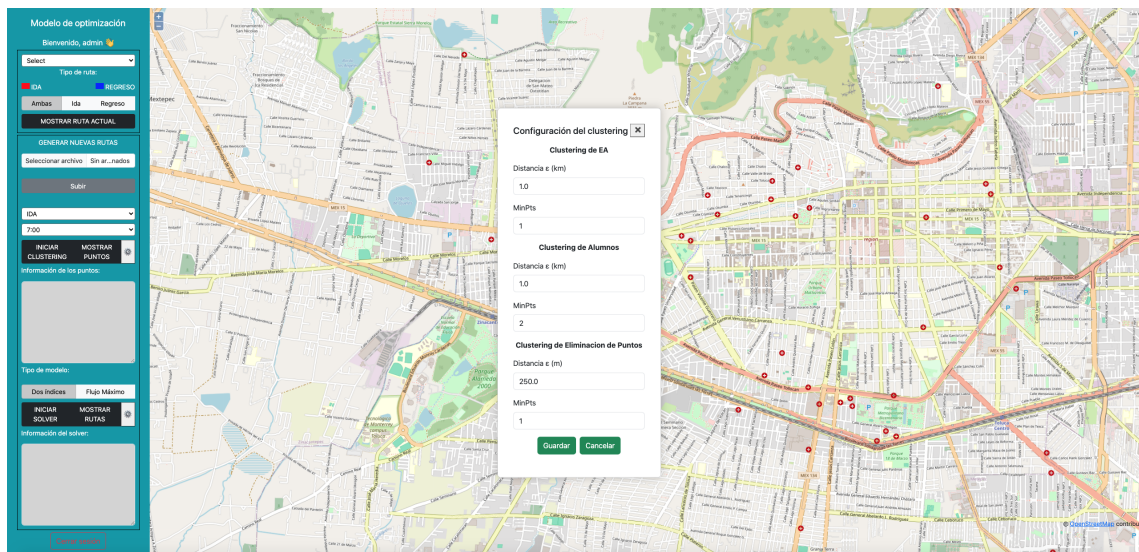


Figura 5.24: Configuraciones para el clustering (Elaboración propia).

En la imagen de la Figura 5.25 se muestran los elementos a configurar para la etapa de la generación de rutas, este proceso se basan en dos categorías: configuraciones del solver CBC y los parámetros del modelo matemático. Los elementos permitidos a configurar para el solver, consisten en el máximo número de soluciones a esperar; así como el tiempo máximo de ejecución, medido en segundos, para que el solver CBC encuentre una solución entera del modelo. Por otro lado, los parámetros a configurar para el modelo consisten primeramente en los coeficientes  $\alpha$ ,  $\beta$  y  $\gamma$ , que permiten dar más prioridad a los elementos de la función objetivo, descritos en la sección 5.3.3 en la Ecuación 5.28. Los otros parámetros permitidos a configurar consisten en la velocidad del vehículo (medido en km/hr), capacidad y cantidad de vehículos, hora de llegada al destino, hora de salida mínima y el tiempo de espera en las paradas de autobús.

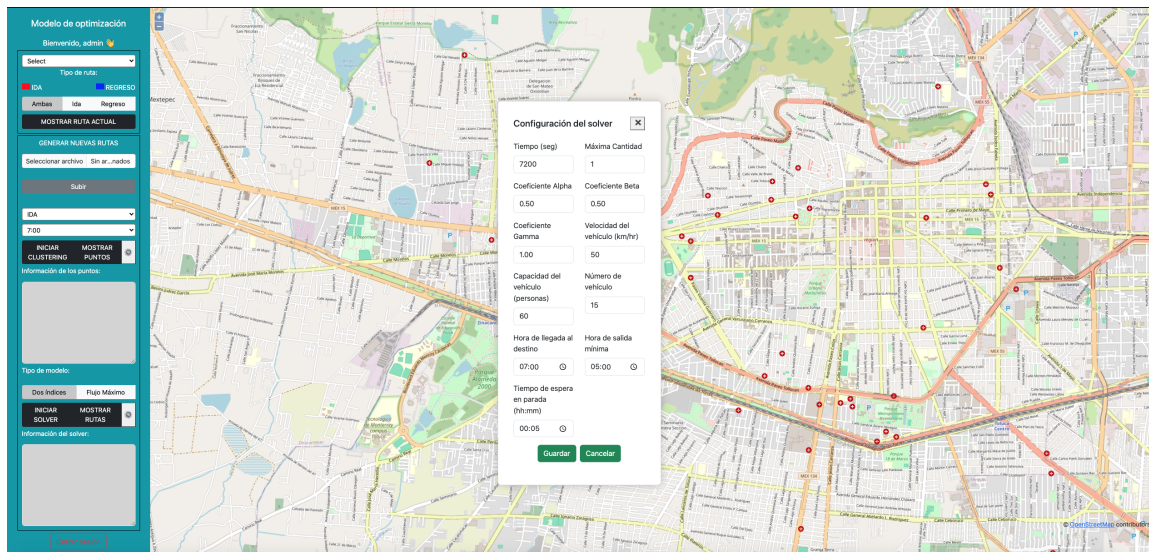


Figura 5.25: Configuraciones para el solver (Elaboración propia).

Si bien el prototipo es funcional, su desarrollo bajo un entorno web, tiene ciertas limitaciones como el requerir de un servidor web para su correcto funcionamiento, lo que repercute en el rendimiento del sistema al resolver el modelo para generar rutas; sin embargo, mantienen una funcionalidad y estabilidad adecuada con los lenguajes y herramientas utilizadas para su desarrollo. Para las etapas de preprocesamiento de datos y de los procesos de clustering, el prototipo funciona correctamente, permitiendo un análisis rápido de los resultados mediante un mapa y marcadores.

En la sección 6.2 del capítulo 6, se hace mención de algunas mejoras que se pueden hacer hacia el prototipo propuesto, así como de la mejora a los procesos de solución del modelo.

---

# Capítulo 6

## Conclusiones y trabajo futuro

---

En este capítulo se exponen todas las conclusiones del análisis del proceso de experimentación, realizado en el capítulo 5, a las cuales se han podido llegar dada la metodología establecida en la sección 1.5 del capítulo 1; además se presenta una sección dedicada a las mejoras posibles tomando como base este trabajo.

### 6.1. Conclusiones

En esta tesis se ha abordado un caso de estudio, descrito con detalle en la sección 1.1, que en resumen plantea generar un nuevo conjunto de rutas de transporte para el traslado de estudiantes mediante la determinación de nuevos puntos de paradas para hasta 45 autobuses, por lo que se utilizaron varias etapas de clustering y un modelo ILP con flujo máximo.

Con las pruebas realizadas al modelo propuesto, se puede decir que tanto la meta de ingeniería y el objetivo general, planteados en la sección 1.3 y 1.4.1 respectivamente, se ha alcanzado ya que las rutas obtenidas llevan la mayor carga de alumnos sin tener un excedente de pasajeros, además de utilizar horarios para garantizar la llegada de los estudiantes en el tiempo marcado. Respecto al compendio de objetivos específicos, establecidos en la sección 1.4.2, se lograron cumplir los primeros 5 objetivos y también el séptimo; sin embargo, culminar el sexto objetivo, que implica la comparativa de las rutas actuales con las nuevas rutas generadas por el modelo mediante la función objetivo, no fue completamente viable ya que se requiere la totalidad de las rutas para los 45 vehículos, es decir, es necesario la solución del escenario 3. En los siguientes párrafos se menciona con más detalle cada una de las conclusiones y hallazgos del proceso realizado en este trabajo.

Los procesos de clustering aplicados a la matrícula de alumnos, mediante el uso de códigos postales, determinó adecuadamente los puntos de paradas para la recolección de alumnos, permitiendo reducir la cantidad de datos significativamente y sin pérdida, mediante

la distinción de *destinos* (a los cuales los alumnos se dirigen); sin embargo, la precisión de los puntos de parada para las rutas de los vehículos puede ser mejorada con la adición de campos de dirección más precios, como lo es el uso de colonias.

Con los elementos obtenidos por los procesos descritos, se pudieron establecer los subconjuntos de vértices necesarios (paradas, destinos y terminales) para el modelo de ILP, que se basó en el conocido Problema de Enrutamiento de Vehículos, con la cualidad de movilizar un flujo de estudiantes mediante los componentes que presenta el Problema de Flujo Máximo en redes.

Al separar y modelar las rutas en sentido de IDA y REGRESO, permitió generar un modelo de ILP más eficiente que obtuvo soluciones en un menor tiempo de ejecución, en comparación de modelos convencionales de CVRP, esto fue debido principalmente a las restricciones de eliminación de subciclos basadas en la restricción MTZ. Sin embargo, para el modelo que utiliza flujo máximo el tener una variable de decisión con cuatro índices implicó que para el escenario de prueba 3, se tuviese un número de variables de decisión elevado, mostrando que con el incremento de nodos y destino la resolución de este conlleva mucho tiempo computacional. Esto puede ser estimado tomando a  $n$  como el número de nodos,  $m$  como la cantidad de vehículos disponibles y al elemento  $d$  como la cantidad de destinos; por lo que la complejidad de los datos puede ser denotada como  $O(n^2 \cdot md)$ . Como los modelos propuestos parten de VRP y este a su vez parte del TSP, siendo este último un problema NP-hard, la complejidad del modelo trabajado también se encuentra dentro de esta categoría.

A pesar de la complejidad del modelo a flujo máximo, la solución de éste, permite obtener rutas de transporte para cada vehículo evitando la saturación de usuarios, en comparativa a las rutas actuales de transporte, además de dar la posibilidad de llevar la mayor cantidad de usuarios a un costo mínimo (tiempo) y garantizando que las rutas por vehículo lleven a los usuarios a los destinos que les corresponden. La realización de una comparativa más completa, entre las rutas actuales y las generadas por el modelo de flujo máximo, mediante la función objetivo no fue viable al no tener el compendio de rutas por vehículo del escenario 3. Sin embargo, esto puede ser solventado mediante la solución del modelo para el escenario 3 mediante algún método aproximado (heurístico o metaheurístico) y de esta manera utilizar la función objetivo como punto de comparación.

Otra alternativa para el escenario 3, es tener una formulación más sencilla para el modelo presentado en la sección 5.3.3. Para esta situación está la opción de definir una gráfica que represente los caminos y modelar el comportamiento de flujo sobre las aristas ( $e \in A$ ) en lugar de los vértices ( $i, j \in V, i \neq j$ ), como se ha estado realizando. Esto se logra mediante una función  $\delta(i)$  que retorna las aristas correspondientes al vértice  $i$ [6].

Con lo mencionado hasta el momento, las rutas obtenidas en los escenarios de pruebas, evitan tener puntos innecesarios de descenso y garantizando el traslado de la mayor cantidad de alumnos hacia sus espacios académicos correspondientes, permitiendo que las rutas lleguen conforme los horarios de entrada y teniendo tiempos de espera, por parada, adecuados.

## 6.2. Trabajo Futuro

Como trabajo futuro se propone mejorar el modelo de flujo máximo, mediante algunos elementos del área de teoría de gráficas, como los árboles generadores mínimos (MST, por sus siglas en inglés), para otorgarle al modelo de ILP la característica de poder generar rutas para cada vehículo que vayan a más de un destino y aprovechar de mejor manera el flujo máximo a costo mínimo que otorga el modelo propuesto. Otro componente interesante a trabajar, es el desarrollo de una metaheurística, que tome de base el modelo matemático propuesto, que permita obtener soluciones adecuadas en tiempos computacionales cortos, para instancias superiores a los 50 nodos con más de tres destinos y más de 15 vehículos; siendo necesario el desarrollo de operadores y representaciones adecuados para problemas combinatorios.

El modelo propuesto de la sección 5.3.3, puede ser formulado con solo tres índices, si ya se tienen una gráfica predefinida; es decir, ya teniendo una red se puede insertar flujo sobre dicha red, por lo que otro punto de interés para esta investigación es adaptar el modelo propuesto a una gráfica ya dirigida y establecida, para analizar el comportamiento de las rutas al añadir el concepto de flujo máximo.

Por otra parte, el prototipo de sistema puede ser mejorado en muchos aspectos, desde los lenguajes utilizados, hasta incorporar frameworks conocidos para un mejor desarrollo del sistema, de igual manera, se puede cambiar los solvers gratuitos por comerciales cuya eficiencia es mejor, entre los que se pueden mencionar están: Gurobi y CPLEX.

Las contribuciones del presente trabajo, a la literatura existente, son primeramente un modelo de ILP que permite generar rutas de transporte que permiten llevar la mayor cantidad de estudiantes hacia sus destinos correspondientes a un costo mínimo (tiempo), a diferencia de los modelos existentes de SBRP (School Bus Routing Problem); además se utilizan procesos de clustering y una envolvente convexa que determinan un nuevo conjunto de paradas de autobús basadas en la distribución geográfica de los estudiantes, lo cual conlleva a determinar de manera adecuada los parámetros y conjuntos necesarios para alimentar al modelo de ILP y obtener rutas de transporte óptimas y factibles.



# Referencias

---

- [1] L. Ruisong and W. Ning, “Data-driven bus route optimization algorithm under sudden interruption of public transport,” *IEEE Access*, vol. 10, pp. 5250–5263, 2022, doi:10.1109/ACCESS.2022.3140947.
- [2] Banco Mundial. (2023) Transporte: Panorama general. [Online]. Available: <https://www.bancomundial.org/es/topic/transport/overview>
- [3] Y. Li and S. Lu, “Study on the optimization of urban passenger traffic structure based on multi-objective linear programming—a case study of beijing,” *Environmental Science and Pollution Research*, vol. 28, no. 8, pp. 10 192–10 206, 2021, doi:10.1007/s11356-020-11358-y.
- [4] K. A. Tütüncü, N. N. Gül, U. Bölükbaş, and A. F. Güneri, “Integer linear programming approach for the personnel shuttles routing problem in yıldız campus in istanbul,” *Journal of Soft Computing and Decision Analytics*, vol. 1, no. 1, pp. 303—316, Oct. 2023, doi:10.31181/jscda11202326.
- [5] R. H. Ballou, *Lógica de administración de la cadena de suministro*, 5th ed. México: Pearson Educación, 2010.
- [6] P. Toth and D. Vigo, Eds., *The vehicle routing problem*, ser. SIAM monographs on discrete mathematics and applications. Philadelphia, Pa: Society for Industrial and Applied Mathematics, 2002.
- [7] M. A. Sánchez, “Algoritmos de clustering para la detección de posibles paradas de autobús,” *Informaticae Abstracta*, vol. 2, no. 1, pp. 4–24, 2024. [Online]. Available: <https://informaticae.uaemex.mx/article/view/24595>
- [8] M. A. Sánchez and L. Loza, “Envoltura convexa para determinar puntos terminales en problemas de enrutamiento de vehículos capacitados,” *Informaticae Abstracta*, vol. 2, no. 2, pp. 60–74, 2025. [Online]. Available: <https://informaticae.uaemex.mx/article/view/25309>

- [9] datascientest. (2023) Machine learning y clustering: el algoritmo dbscan. Consultado el 25 de noviembre de 2023. [Online]. Available: <https://datascientest.com/es/machine-learning-clustering-dbscan>
- [10] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 1st ed. O’Reilly Media, 2019.
- [11] Departamento de Matemáticas e Informática de la Universidad de Barcelona, *El poder de los datos, Del big data al aprendizaje profundo*. National Geographic, 2017.
- [12] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatian databases with npise,” in *Proc. AAAI*, vol. 23, no. 1, pp. 226–231, 1996.
- [13] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, “DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN,” *ACM Transactions on Database Systems*, vol. 42, no. 3, pp. 1–21, Sep. 2017, doi:10.1145/3068335.
- [14] P. Grabusts, “The Choice of Metrics for Clustering Algorithms,” *Environment. Technology. Resources. Proceedings of the International Scientific and Practical Conference*, vol. 2, p. 70, Aug. 2015, doi:10.17770/etr2011vol2.973.
- [15] W. Zhang, “Applications of the haversine formula in the evaluation of the great-circle distance between two points on the earth’s surface,” *Journal of Applied Mathematics and Physics*, vol. 3, no. 7, pp. 812–819, 2009.
- [16] P. Dauni, M. D. Firdaus, R. Asfariani, M. I. N. Saputra, A. A. Hidayat, and W. B. Zulfikar, “Implementation of haversine formula for school location tracking,” *Journal of Physics: Conference Series*, vol. 1402, no. 7, p. 077028, 2019, doi:10.1088/1742-6596/1402/7/077028.
- [17] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Prentice Hall PTR, 1996.
- [18] M. Halkidi, “On Clustering Validation Techniques,” *Journal of Intelligent Information Systems*, vol. 17, pp. 107–145, 2001, doi:10.1023/A:1012801612483.
- [19] J. Bezdek and N. Pal, “Cluster validation with generalized dunn’s indices,” in *Proceedings 1995 Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems*, 1995, pp. 190–193, doi:10.1109/ANNES.1995.499469.
- [20] scikit-learn developers. (2024) Clustering — scikit-learn documentation. Accessed: Jul. 04, 2024. [Online]. Available: <https://scikit-learn.org/stable/modules/clustering.html>

- [21] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, Nov. 1987, doi:10.1016/0377-0427(87)90125-7.
- [22] D. L. Davies and D. W. Bouldin, “A Cluster Separation Measure,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979, doi:10.1109/TPAMI.1979.4766909.
- [23] T. Calinski and J. Harabasz, “A dendrite method for cluster analysis,” *Communications in Statistics - Theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974, doi:10.1080/03610927408827101.
- [24] R. K. Cersonsky and S. De, “Chapter 7 - unsupervised learning,” in *Quantum Chemistry in the Age of Machine Learning*, P. O. Dral, Ed. Elsevier, 2023, pp. 153–181, doi:10.1016/B978-0-323-90049-2.00025-1.
- [25] M. Misuraca, M. Spano, and S. Balbi, “Bms: An improved dunn index for document clustering validation,” *Communications in Statistics - Theory and Methods*, vol. 48, no. 20, pp. 5036–5049, 2019, doi:10.1080/03610926.2018.1504968.
- [26] J. Ohser and F. Mücklich, *Statistical Analysis of Microstructures in Materials Science*. John Wiley & Sons, Ltd, 2000.
- [27] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*, 3rd ed. Springer, 2008, capítulo 1, Envolvente convexa.
- [28] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986, capítulo 3, Estimación de densidad de Kernel.
- [29] J. Nocedal and S. J. Wright, *Numerical optimization*, 2nd ed., ser. Springer series in operation research and financial engineering. New York, NY: Springer, 2006.
- [30] H. A. Taha, *Investigación de operaciones*, 9th ed. Mexico, D.F.: Pearson Educación, 2015.
- [31] Sociedad Mexicana de Investigación de Operaciones. (2023) Investigación de operaciones. [Online]. Available: <http://www.smio.org>
- [32] B. Chopard and M. Tomassini, *An Introduction to Metaheuristics for Optimization*, ser. Natural Computing Series. Cham: Springer International Publishing, 2018, doi:10.1007/978-3-319-93073-2.
- [33] R. D. Carr and N. Simonetti, “A new integer programming formulation of the graphical traveling salesman problem.” [Online]. Available: <http://arxiv.org/abs/2006.04933>

- [34] G. B. Dantzig and J. H. Ramser, “The Truck Dispatching Problem,” *Management Science*, vol. 6, no. 1, pp. 80–91, 1959. [Online]. Available: <http://www.jstor.org/stable/2627477>
- [35] J. A. Bondy and U. S. R. Murty, *Graph Theory*, ser. Graduate Texts in Mathematics, S. Axler and K. A. Ribet, Eds. London: Springer London, 2008, vol. 244.
- [36] AIMMS B.V. (2023) Time windows. [Online]. Available: <https://how-to.aimms.com/Articles/332/332-Time-Windows.html>
- [37] M. Sciortino, R. Lewis, and J. Thompson, “A school bus routing heuristic algorithm allowing heterogeneous fleets and bus stop selection,” *SN Computer Science*, vol. 4, no. 1, p. 74, 2022, doi:10.1007/s42979-022-01466-6.
- [38] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows. Theory, Algorithms and Applications*. Prentice Hall, 1993.
- [39] G. B. Dantzig, “Application of the simplex method to a transportation problem,” *Activity Analysis of Production and Allocation*, vol. 13, pp. 359–373, 1951.
- [40] JuMP Development Team, “Documentación de jump.jl (versión estable),” <https://jump.dev/JuMP.jl/stable/>, 2025, accedido el 23 de junio de 2025. [Online]. Available: <https://jump.dev/JuMP.jl/stable/>
- [41] J. Forrest and R. Lougee-Heimer, “CBC User Guide,” in *Emerging Theory, Methods, and Applications*, H. J. Greenberg and J. C. Smith, Eds. INFORMS, Sep. 2005, pp. 257–277, doi:10.1287/educ.1053.0020.
- [42] A. Makhorin, *GNU Linear Programming Kit (GLPK)*, 2024, version 5.0. [Online]. Available: <https://www.gnu.org/software/glpk/>
- [43] L. Gurobi Optimization, *Gurobi Optimizer Reference Manual*, 2024. [Online]. Available: <https://www.gurobi.com>
- [44] I. Corporation, *IBM ILOG CPLEX Optimization Studio CPLEX User’s Manual*, 2024. [Online]. Available: <https://www.ibm.com/products/ilog-cplex-optimization-studio>
- [45] C. S. W. and K. M. R., “Optimisation of vehicle routing problem with time windows using harris hawks optimiser,” *Journal of Mechanical Engineering and Sciences*, vol. 16, no. 39, pp. 9056–9065, 2022, doi:10.15282/jmes.16.3.2022.08.0717.
- [46] F. Glover and B. Melián, “Búsqueda tabú,” *Revista Iberoamericana de Inteligencia Artificial*, vol. 24, no. 19, pp. 29–48, 2003.

- [47] Z. Jun and Z. Jing, “Vehicle routing problems with time windows based on the improved hybrid fish swarm-ant colony algorithm,” *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 2022, doi:10.1007/s12008-022-00953-7.
- [48] B. Moradi, “The new optimization algorithm for the vehicle routing problem with time windows using multi-objective discrete learnable evolution model,” *Soft Computing*, vol. 24, no. 9, pp. 6741–6769, 2020, doi:10.1007/s00500-019-04312-9.
- [49] H.-B. Ban, H.-P. Nguyen, and D.-H. Pham, “Hybrid variable neighborhood search for solving school bus-driver problem with resource constraints,” *Computer Science*, vol. 24, no. 3, pp. 297–320, 2023, doi:10.7494/csci.2023.24.3.4367.
- [50] H. Zhang, H. Ge, J. Yang, and Y. Tong, “Review of Vehicle Routing Problems: Models, Classification and Solving Algorithms,” *Archives of Computational Methods in Engineering*, vol. 29, no. 1, pp. 195–221, Jan. 2022, doi:10.1007/s11831-021-09574-x.
- [51] J. R. Montoya-Torres, J. López Franco, S. Nieto Isaza, H. Felizzola Jiménez, and N. Herazo-Padilla, “A literature review on the vehicle routing problem with multiple depots,” *Computers & Industrial Engineering*, vol. 79, pp. 115–129, Jan. 2015, doi:10.1016/j.cie.2014.10.029.
- [52] N. A. El-Sherbeny, “Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods,” *Journal of King Saud University - Science*, vol. 22, no. 3, pp. 123–131, Jul. 2010, doi:10.1016/j.jksus.2010.03.002.
- [53] A. Villalba and E. C. Gonzalez la Rotta, “Comparison of dbscan and k-means clustering methods in the selection of representative clients for a vehicle routing model,” in *2020 Congreso Internacional de Innovación y Tendencias en Ingeniería (CONIITI)*, 09 2020, pp. 1–6, doi:10.1109/CONIITI51147.2020.9240399.
- [54] S. S. Néia, A. O. Artero, and C. B. D. Cunha, “QUALITY ANALYSIS FOR THE VRP SOLUTIONS USING COMPUTER VISION TECHNIQUES,” *Pesquisa Operacional*, vol. 37, no. 2, pp. 387–402, Aug. 2017, doi:10.1590/0101-7438.2017.037.02.0387.
- [55] B. Kallehauge, “Formulations and exact algorithms for the vehicle routing problem with time windows,” *Computers & Operations Research*, vol. 35, no. 7, pp. 2307–2330, Jul. 2008, doi:10.1016/j.cor.2006.11.006.
- [56] T. Ralphs, L. Kopman, W. Pulleyblank, and L. Trotter, “On the capacitated vehicle routing problem,” *Mathematical Programming*, vol. 94, no. 2-3, pp. 343–359, Jan. 2003, doi:10.1007/s10107-002-0323-0.
- [57] J. M. Belenguer, M. C. Martinez, and E. Mota, “A Lower Bound for the Split Delivery Vehicle Routing Problem,” *Operations Research*, vol. 48, no. 5, pp. 801–810, Oct. 2000, doi:10.1287/opre.48.5.801.12407.

- [58] H. Tang and E. Miller-Hooks, “Interactive Heuristic for Practical Vehicle Routing Problem with Solution Shape Constraints,” *Transportation Research Record*, vol. 1964, no. 1, pp. 9–18, 2006, doi:10.1177/0361198106196400102.
- [59] M. M. Alipour and S. N. Razavi, “A new local search heuristic based on nearest insertion into the convex hull for solving Euclidean TSP,” *International Journal of Operational Research*, vol. 34, pp. 409–429, 2019, doi:10.1504/IJOR.2019.10019739.
- [60] Openstreetmap.org. (2024) Openstreetmap. Consultado el 21 de mayo de 2024. [Online]. Available: <https://www.openstreetmap.org>
- [61] C. P. Mexicano. (2023) Servicio postal mexicano. Consultado el 12 de noviembre de 2023. [Online]. Available: <https://www.correosdemexico.gob.mx/>
- [62] V. Belton and T. J. Stewart, *Multiple Criteria Decision Analysis: An Integrated Approach*. Boston, MA: Springer, 2002.