

Pontificia Universidad Católica del Ecuador

Facultad De Ingeniería

Escuela de Sistemas



TEMA:

Automatización de la logística de última milla mediante técnicas de inteligencia artificial

AUTOR:

Sebastián Felipe Tamayo Proaño

TRABAJO PREVIA A LA OBTENCIÓN DEL TÍTULO DE MAGÍSTER EN SISTEMAS DE
INFORMACIÓN CON MENCIÓN EN DATA SCIENCE

QUITO, 2024 – 2025

DEDICATORIA

En primer lugar, quiero dedicar este trabajo a mi difunto gato Tomás, el poco tiempo que formo parte de mi vida fue muy gratificante, su partida no será olvidada.

Quiero dedicar también este trabajo a mi persona debido a que durante todo este proceso han habido momentos de debilidad en los que he considerado rendirme pero gracias a Dios, mi familia y a mí mismo, he logrado sobreponerme a todos los obstáculos, por lo que, quiero aprovechar este momento para dedicar este logro a mi yo del futuro, en especial cuando me enfrente a obstáculos que me hagan pensar que no hay solución, ten fe Sebastián, confía en ti, en Dios y en tu familia, siempre se buena persona y veras todo saldrá bien SIEMPRE.

Finalmente, quiero dedicar este trabajo a todas aquellas personas y mascotas que he dejado de lado en el camino para finalizar este trabajo, es verdad que tal vez he perdido momentos especiales a su lado, pero les aseguro todo ha valido la pena, espero que este trabajo sea de su agrado si se animan a leerlo.

AGRADECIMIENTO

En primer lugar, quiero agradecer a Dios por darme la fortaleza y sabiduría para poder completar este trabajo que corresponde a un gran esfuerzo y dedicación que he invertido solo con el objetivo de honrarlo, gracias por siempre estar y escucharme en mis momentos más débiles, este trabajo y mi vida son para honrar tu palabra y hacer el bien.

Adicionalmente, quiero agradecer a mi familia, en especial a mi padre Fabricio Tamayo, mi madre Cristina Proaño, mi hermana Daniela Tamayo, mis abuelitos Ramiro Tamayo, Mercedes Proaño, Ana Fierro y mi difunto abuelito José Proaño, y a mis gatos Kirk, Lara, Coquita, Lara y mi difunto gato Tomás; si no fuera por su apoyo incondicional, no hubiera podido completar este trabajo de titulación, estoy muy agradecido por todo lo que hacen por mí y espero, de todo corazón, corresponderles.

Finalmente, quisiera agradecer a la Pontificia Universidad Católica del Ecuador por ayudarme a crecer profesional y personalmente, considero que es una muy buena universidad y tras todos los años que he pertenecido a esta institución desde grado hasta esta maestría, considero que puedo decir que he aprendido mucho y estoy orgulloso de formar parte de esta honorable institución.

RESUMEN

El presente trabajo de titulación corresponde al desarrollo de LUMA (Logística de Última Milla Automatizada) cuyo enfoque consiste en la automatización de la fase de planificación de la asignación de mercancías a vehículos de despacho.

A lo largo de cada capítulo, se detallan los procesos que fueron empleados para el desarrollo de LUMA, entre ellos, se utilizaron técnicas de aprendizaje no supervisado mediante la agrupación en clústeres que permitió identificar sectores de entrega a los que, posteriormente, se les asignó vehículos de despacho de acuerdo con un algoritmo que fue desarrollado para dicho propósito; para que, finalmente, las mercancías identificadas de cada sector, sean atribuidas a los vehículos correspondientes.

Sin embargo, para esto se tuvo que realizar una exploración y preparación de datos con registros históricos de mercancías despachadas, donde, se emplearon diversas técnicas para identificar parámetros óptimos que permitieron obtener un resultado adecuado del modelo de agrupación en clústeres.

Finalmente, se demostró que la propuesta de solución para la optimización de la logística de última milla, basada en el uso de aprendizaje no supervisado y la agrupación en clústeres, tuvo éxito al disminuir los porcentajes de subutilización de vehículos en cuanto a sus capacidades.

ÍNDICE

| | |
|--|-----|
| ÍNDICE DE FIGURAS, GRÁFICOS Y TABLAS | VI |
| ÍNDICE DE FIGURAS..... | VI |
| ÍNDICE DE TABLAS | XII |
| CAPÍTULO I: INTRODUCCIÓN..... | 1 |
| 1. Introducción | 1 |
| 1.1. Justificación | 1 |
| 1.2. Planteamiento del problema..... | 3 |
| 1.3. Objetivos..... | 4 |
| 1.3.1. Objetivo General..... | 4 |
| 1.3.2. Objetivos Específicos | 4 |
| 1.4. Alcance | 5 |
| CAPÍTULO II: REVISIÓN SISTEMÁTICA DE LITERATURA | 6 |
| 2. Revisión sistemática de literatura | 6 |
| 2.1. Logística de la Última Milla | 8 |
| 2.2. Inteligencia Artificial y la optimización de la logística de la Última Milla..... | 11 |
| 2.3. El Vehicle Routing Problem (VRP) y la logística de la Última Milla..... | 12 |
| CAPÍTULO III: MARCO TEÓRICO..... | 15 |
| 3. Marco Teórico..... | 15 |

| | |
|--|----|
| 3.1. CRISP-DM..... | 15 |
| 3.1.1. Entendimiento del negocio | 16 |
| 3.1.2. Entendimiento de los datos..... | 17 |
| 3.1.3. Preparación de los datos | 18 |
| 3.1.4. Modelado | 18 |
| 3.1.5. Evaluación | 19 |
| 3.1.6. Despliegue | 19 |
| 3.2. Aprendizaje automático no supervisado | 20 |
| 3.3. Agrupación en clústeres | 20 |
| CAPÍTULO IV: ENTENDIMIENTO DEL NEGOCIO..... | 21 |
| 4. Entendimiento del negocio | 21 |
| 4.1. Objetivos del negocio | 21 |
| 4.1.1. Antecedentes..... | 21 |
| 4.1.2. Objetivo del Negocio..... | 23 |
| 4.1.3. Criterios de Éxito..... | 23 |
| 4.2. Evaluación de la situación | 23 |
| 4.2.1. Inventario de Recursos | 23 |
| 4.2.2. Requerimientos, Supuestos y Limitaciones..... | 24 |
| 4.2.3. Terminología | 25 |
| 4.3. Objetivos de minería de datos..... | 26 |

| | |
|--|----|
| 4.3.1. Objetivos de minería de datos | 26 |
| 4.4. Plan del proyecto..... | 27 |
| 4.4.1. Interpretación del problema de ruteo de vehículos (VRP)..... | 27 |
| 4.4.2. Propuesta de solución del problema y de desarrollo de LUMA..... | 27 |
| CAPÍTULO V: ENTENDIMIENTO DE LOS DATOS..... | 29 |
| 5. Entendimiento de los datos | 29 |
| 5.1. Recopilación de datos | 29 |
| 5.1.1. Descripción de los datos requeridos | 29 |
| 5.1.2. Resumen de los datos recolectados | 30 |
| 5.2. Descripción de los datos | 30 |
| 5.2.1. Descripción del conjunto de datos de Documentos..... | 31 |
| 5.2.2. Descripción del conjuntos de datos de Vehículos | 37 |
| 5.3. Exploración de los datos | 39 |
| 5.3.1. Hallazgos de la exploración de los datos de Documentos..... | 39 |
| 5.3.2. Hallazgos de la exploración de los datos de Vehículos..... | 46 |
| 5.3.3. Hallazgos de la exploración de los datos integrados | 48 |
| 5.4. Verificación de la calidad de los datos | 57 |
| CAPÍTULO VI: PREPARACIÓN DE DATOS..... | 62 |
| 6. Preparación de datos | 62 |
| 6.1. Identificación de variables para la selección de datos | 62 |

| | |
|--|------------|
| 6.2. Selección de variables | 66 |
| 6.3. Resumen de variables que permanecerán | 68 |
| 6.4. Limpieza de datos | 70 |
| 6.5. Formateo de variables | 75 |
| 6.6. Estructuración de los conjuntos de datos | 77 |
| 6.7. Resultados de la preparación de datos | 77 |
| CAPÍTULO VII: DESARROLLO DE LUMA..... | 80 |
| 7. Desarrollo de LUMA | 80 |
| 7.1. Selección de variables que serán utilizadas por el modelo de agrupación en clústeres | 80 |
| 7.2. Identificación de parámetros óptimos | 81 |
| 7.3. Agrupación en clústeres mediante KMeans..... | 83 |
| 7.4. Algoritmos de asignación de vehículos a los sectores identificados y posterior asignación de documentos a dichos vehículos | 84 |
| CAPÍTULO VIII: EVALUACIÓN..... | 95 |
| 8. Evaluación..... | 95 |
| CAPÍTULO IX: CONCLUSIONES Y RECOMENDACIONES | 106 |
| 9. Conclusiones y recomendaciones | 106 |
| 9.1. Conclusiones | 106 |
| 9.2. Recomendaciones | 107 |

BIBLIOGRAFÍA 108

ÍNDICE DE FIGURAS, GRÁFICOS Y TABLAS

ÍNDICE DE FIGURAS

| | |
|--|----|
| Ilustración 1 Ciclo de vida de la minería de datos | 16 |
| Ilustración 2 Diagrama de flujo del proceso de logística de última milla actual | 22 |
| Ilustración 3 Carga del conjunto de datos de Documentos | 31 |
| Ilustración 4 Dimensiones del conjunto de datos de Documentos | 31 |
| Ilustración 5 Carga del conjunto de datos de Vehículos | 37 |
| Ilustración 6 Dimensiones del conjunto de datos de Vehículos | 37 |
| Ilustración 7 Distribución de documentos por su estado..... | 40 |
| Ilustración 8 Estadísticas descriptivas de la cantidad de documentos que se emiten por día..... | 42 |
| Ilustración 9 Cantidad de documentos emitidos para su despacho por día | 43 |
| Ilustración 10 Cantidad de documentos emitidos para su despacho y el peso total por despachar por día | 44 |
| Ilustración 11 Cantidad de documentos emitidos para su despacho y el volumen total por despachar por día | 44 |
| Ilustración 12 Listado de variables con valores constantes..... | 45 |
| Ilustración 13 Estadísticas descriptivas de variables con valores constantes..... | 46 |
| Ilustración 14 Estadísticas descriptivas de las variables con valores constantes del conjunto de datos de Vehículos | 47 |
| Ilustración 15 Visualización de las categorías de la variable <i>ultimoAnioMatricula</i> | 47 |
| Ilustración 16 Función para la creación de variables identificadoras | 48 |

| | |
|---|----|
| Ilustración 17 Identificación de variables requeridas para la creación de variables identificadoras en el conjunto de datos de Documentos | 49 |
| Ilustración 18 Creación de variables identificadores y eliminación de variables usadas para ello..... | 50 |
| Ilustración 19 Capacidad máxima de peso de la flota de vehículos de la empresa en cuestión en relación con el total de peso por despachar por día | 51 |
| Ilustración 20 Capacidad máxima de volumen de la flota de vehículos de la empresa en cuestión en relación con el total de volumen por despachar por día | 52 |
| Ilustración 21 Distribución de pesos asignados a vehículos | 53 |
| Ilustración 22 Distribución de volúmenes asignados por vehículo | 53 |
| Ilustración 23 Rangos de porcentajes de subutilización de vehículos con respecto al peso | 54 |
| Ilustración 24 Rangos de porcentajes de subutilización de vehículos con respecto al volumen..... | 55 |
| Ilustración 25 Capacidad máxima del vehículo con mayor capacidad con respecto al peso en relación con el documento con mayor peso por despachar por día | 56 |
| Ilustración 26 Capacidad máxima del vehículo con mayor capacidad con respecto al volumen en relación con el documento con mayor volumen por despachar por día | 56 |
| Ilustración 27 Distribución de valores faltantes por variable..... | 57 |
| Ilustración 28 Estadísticas descriptivas de variables con valores constantes del conjunto de datos de Documentos | 59 |
| Ilustración 29 Estadísticas descriptivas de las variables con valores constantes del conjunto de datos de Vehículos | 59 |

| | |
|---|----|
| Ilustración 30 Proceso de descarte de variables no necesarias del conjunto de datos de Documentos | 67 |
| Ilustración 31 Proceso de descarte de variables no necesarias del conjunto de datos de Vehículos | 67 |
| Ilustración 32 Vista de las nuevas dimensiones de los conjuntos de datos | 68 |
| Ilustración 33 Valores faltantes por variable del conjunto de datos de Documentos..... | 70 |
| Ilustración 34 Verificación de documentos con coordenadas erróneas | 71 |
| Ilustración 35 Eliminación de valores faltantes en el conjunto de datos de Documentos y reporte de sus resultados | 72 |
| Ilustración 36 Procesamiento de documentos erróneos | 72 |
| Ilustración 37 Identificación de documentos anormales | 73 |
| Ilustración 38 Evaluación de igualdad entre los casos de anormalidad presentados en los documentos | 74 |
| Ilustración 39 Eliminación de documentos anormales y reporte de sus resultados | 75 |
| Ilustración 40 Formateo de la variable <i>estadoDocumento</i> e <i>id_vehiculo</i> del conjunto de datos de Documentos | 76 |
| Ilustración 41 Estructuración del orden de las variables de los conjuntos de datos de Documentos y Vehículos | 77 |
| Ilustración 42 Visualización del conjunto de datos de Documentos preparado..... | 78 |
| Ilustración 43 Visualización del conjunto de datos de Vehículos preparado..... | 79 |
| Ilustración 44 Código para la selección de variables que serán utilizadas por el modelo de agrupación en clústeres | 80 |
| Ilustración 45 Función que permite la identificación del parámetro k_max | 82 |

| | |
|---|----|
| Ilustración 46 Función que permite la identificación del valor de k óptimo | 83 |
| Ilustración 47 Agrupación en clústeres, modelo, entrenamiento y asignación | 84 |
| Ilustración 48 Creación de un dataframe con información de los sectores identificados | 85 |
| Ilustración 49 Ilustración 1 del algoritmo de asignación de vehículos a los sectores identificados y posterior asignación de documentos a dichos vehículos | 85 |
| Ilustración 50 Ilustración 2 del algoritmo de asignación de vehículos a los sectores identificados y posterior asignación de documentos a dichos vehículos | 86 |
| Ilustración 51 Ilustración 3 del algoritmo de asignación de vehículos a los sectores identificados y posterior asignación de documentos a dichos vehículos | 86 |
| Ilustración 52 Ilustración 4 del algoritmo de asignación de vehículos a los sectores identificados y posterior asignación de documentos a dichos vehículos | 87 |
| Ilustración 53 Ilustración 5 del algoritmo de asignación de vehículos a los sectores identificados y posterior asignación de documentos a dichos vehículos | 87 |
| Ilustración 54 Ilustración 6 del algoritmo de asignación de vehículos a los sectores identificados y posterior asignación de documentos a dichos vehículos | 88 |
| Ilustración 55 Ilustración 7 del algoritmo de asignación de vehículos a los sectores identificados y posterior asignación de documentos a dichos vehículos | 88 |
| Ilustración 56 Algoritmo de asignación inicial de vehículos a los sectores identificados | 90 |
| Ilustración 57 Ilustración 1 del algoritmo de verificación de vehículos asignados con respecto al peso de cada documento | 91 |
| Ilustración 58 Ilustración 2 del algoritmo de verificación de vehículos asignados con respecto al peso de cada documento | 91 |

| | |
|---|-----|
| Ilustración 59 Ilustración 1 del algoritmo de evaluación de necesidad de vehículos asignados..... | 93 |
| Ilustración 60 Ilustración 2 del algoritmo de evaluación de necesidad de vehículos asignados..... | 93 |
| Ilustración 61 Algoritmo de asignación de documentos a los vehículos asignados al sector..... | 94 |
| Ilustración 62 Función que permite obtener los documentos emitidos en una fecha aleatoria..... | 95 |
| Ilustración 63 Obtención de los documentos emitidos en una fecha aleatoria..... | 96 |
| Ilustración 64 Distribución de los puntos de entrega de los documentos emitidos en una fecha aleatoria..... | 96 |
| Ilustración 65 Visualización del dataframe de las variables seleccionadas para la agrupación en clústeres..... | 97 |
| Ilustración 66 Identificación de k_max | 97 |
| Ilustración 67 Identificación del valor de k óptimo | 98 |
| Ilustración 68 Agrupación en clústeres de los documentos emitidos en una fecha aleatoria..... | 98 |
| Ilustración 69 Visualización de los sectores de entrega de los documentos | 99 |
| Ilustración 70 Contextualización del día..... | 99 |
| Ilustración 71 Visualización de los resultados de la agrupación en clústeres y creación de un nuevo dataframe que contenga los datos de cada sector..... | 100 |
| Ilustración 72 Resultados de la visualización del reporte de los sectores de entrega ... | 101 |
| Ilustración 73 Reporte de resultados generales..... | 102 |

| | |
|---|-----|
| Ilustración 74 Reporte de resultados específicos | 102 |
| Ilustración 75 Resultados del reporte específico | 103 |
| Ilustración 76 Reporte de subutilizaciones..... | 103 |
| Ilustración 77 Comparación de porcentajes de subutilización con respecto al peso..... | 104 |
| Ilustración 78 Comparación de porcentajes de subutilización con respecto al volumen | 105 |

ÍNDICE DE TABLAS

| | |
|--|----|
| Tabla 1 Inventario de recursos | 23 |
| Tabla 2 Resumen de los datos recolectados..... | 30 |
| Tabla 3 Variables utilizadas para la identificación de los documentos | 32 |
| Tabla 4 Variables utilizadas para la identificación de los despachadores | 32 |
| Tabla 5 Variables utilizadas para la identificación de los vehículos | 32 |
| Tabla 6 Variables utilizadas para la identificación de los clientes | 33 |
| Tabla 7 Variables utilizadas para la identificación de las direcciones principales de los clientes | 33 |
| Tabla 8 Variables con información geográfica de los clientes | 34 |
| Tabla 9 Variables cronológica de los documentos..... | 34 |
| Tabla 10 Variables con información de los documentos | 34 |
| Tabla 11 Instancias de estados de documentos de la variable <i>estadoDocumento</i> | 35 |
| Tabla 12 Variables con información de la empresa a la que pertenecen los datos | 36 |
| Tabla 13 Variables con información de apoyo | 36 |
| Tabla 14 Variables utilizadas para la identificación de los vehículos | 37 |
| Tabla 15 Variables con información de los vehículos | 38 |
| Tabla 16 Variables con información de la capacidad de los vehículos | 38 |
| Tabla 17 Variables con información de apoyo | 38 |
| Tabla 18 Análisis de variables con valores constantes | 60 |
| Tabla 19 Variables por descartar del conjunto de datos de Documentos | 62 |
| Tabla 20 Variables por descartar del conjunto de datos de Vehículos | 65 |
| Tabla 21 Variables que permanecerán en el conjunto de datos de Documentos | 68 |

Tabla 22 Variables que permanecerán en el conjunto de datos de Vehículos 69

CAPÍTULO I: INTRODUCCIÓN

1. Introducción

1.1. Justificación

Para empezar, Gevaers menciona que “la logística de última milla es el tramo final de un servicio de entrega empresarial mediante el cual el envío se entrega al destinatario” (Gevaers, Van de Voorde, & Vanelslander, 2014), es decir, la logística de la última milla es el movimiento de pedidos desde un centro de distribución hacia sus destinos finales, no obstante, para que esta logística de rutas sea adecuada, no es suficiente con “ser eficientes en cuanto a las millas recorridas, sino que también se debe tener en cuenta variables como la capacidad del vehículo, el tiempo estimado de entrega y mucho más” (DispatchTrack, 2022).

De esta forma, si se planifica la logística de la última milla de manera manual, se podrían elaborar planificaciones relativamente eficaces para rutas simples, sin embargo, cuando aumentan los destinos, restricciones y demás variables propias del proceso, esta planificación podría resultar ineficiente (SimpliRoute, s.f.), por lo que, considerando que las entregas de última milla pueden representar el 50% del coste total de la cadena de suministro (Ousaid, 2020, pág. 24), automatizar la logística de última milla significa mayores ingresos para el negocio.

Además, la necesidad de automatizar de manera adecuada la logística de última milla también recae en que los clientes pueden terminar cancelando sus pedidos si sienten que la empresa se está demorando demasiado en entregar lo que ordenaron, además, ya que los consumidores esperan recibir sus productos a tiempo, las empresas que no cumplen con esta expectativa no lograrán ganarse la lealtad de los clientes (DispatchTrack, 2022), es decir, una adecuada planificación logística es sustancial si se desea no solo mantener a los clientes actuales satisfechos, sino que, de igual manera, podría influenciar en la llegada de nuevos negocios.

De esta manera, para maximizar los ingresos de una empresa y asegurar la lealtad de sus clientes, la automatización, y, por ende, optimización de la logística de la última milla, es imprescindible para que las empresas administren sus flotas de manera eficiente, mejorando sus tiempos de entrega y reduciendo las entregas fallidas.

Por lo que, la inteligencia artificial desempeña un papel crucial ya que, como menciona Jucha, “el uso de inteligencia artificial en la coordinación de la última milla tiene un impacto positivo en su logística” (Jucha, 2021, pág. 5), además, si bien es actualmente aplicada en otras áreas de la logística, existe una falta concreta de desarrollo para la última milla (Rosendorff, Hodes, & Fabian, 2021, pág. 12).

Asimismo, considerando que la inteligencia artificial es capaz de contemplar una amplia gama de variables para asignar vehículos de manera óptima, planificar rutas más eficientes, reducir la cantidad de kilómetros recorridos, y considerar variables logísticas como características de los vehículos y dimensiones de los pedidos (Peralta, 2023), es inevitable que esta tecnología sea suficientemente capaz de superar los métodos de planificación manuales.

Adicionalmente, Llegó menciona que entre los pasos que se pueden seguir para automatizar la logística de la última milla, la utilización de herramientas de aprendizaje automático puede permitir la identificación de patrones y tendencias en la gestión de inventarios y rutas de entrega, lo que permite tomar decisiones más informadas (Llegó, 2023).

Es más, como menciona Patel, algunos de los beneficios de la automatización de la logística de la última milla mediante el uso de aprendizaje automático recaen en un ahorro de costos a partir de distancias de entrega reducidas, mayor satisfacción del cliente con tiempos de entrega adecuados, mayor eficiencia operativa, utilización efectiva de recursos y escalabilidad

para manejar volúmenes de entrega crecientes (Patel, 2024). En otras palabras, el uso y aplicación de aprendizaje automático tiene una gran cantidad de posibles beneficios y ventajas para la planificación de la logística de última milla.

En resumen, la automatización de la logística de última milla mediante técnicas de inteligencia artificial promete superar a los métodos de planificación manuales en diferentes aristas como la asignación de pedidos a vehículos, por lo que, en consecuencia, se pueden obtener beneficios como una mayor eficiencia en la utilización de recursos, una reducción de la cantidad de kilómetros recorridos por vehículo, su tiempo en carretera, e, inclusive, una mejora en la satisfacción de los clientes.

1.2. Planteamiento del problema

La planificación de la logística de última milla llevada a cabo como un proceso manual suele omitir aspectos críticos que yacen en los criterios de selección de los vehículos que serán los encargados de realizar el despacho de las diversas mercancías por ser entregadas, pues, a pesar de que se tome en cuenta la capacidad de los vehículos en cuanto a volumen y peso, no se ha implementado un método por el cual se identifique cuáles o cuántos vehículos son los adecuados para realizar el despacho de las mercancías conferidas a un día y sector específico de manera automática, lo que termina resultando en asignaciones subóptimas de vehículos, es decir, en una distribución ineficiente de recursos.

Con este contexto, se evidencia que el problema principal radica en la falta de identificación de un método efectivo para automatizar la asignación de mercancías a vehículos en la logística de última milla mediante el uso de técnicas de inteligencia artificial; mientras que, los problemas secundarios son:

- No se ha analizado el proceso de asignación de productos a vehículos en la logística de última milla de empresas de distribución.
- No se ha elaborado un plan para la automatización de la logística de última milla mediante técnicas de inteligencia artificial.
- No se ha desarrollado un modelo basado en técnicas de inteligencia artificial que permita automatizar la logística de última milla.

1.3. Objetivos

1.3.1. Objetivo General

Automatizar la asignación de productos a vehículos de despacho en la logística de última milla para empresas de distribución mediante la implementación de técnicas de inteligencia artificial.

1.3.2. Objetivos Específicos

Con base al planteamiento del problema, se pueden definir los siguientes objetivos secundarios:

- Analizar y comprender el proceso de asignación de productos a vehículos en la logística de última milla de empresas de distribución.
- Elaborar un plan para la automatización de la logística de última milla mediante técnicas de inteligencia artificial.
- Desarrollar un modelo basado en la técnica de inteligencia artificial más adecuada para la automatización de la logística de última milla.

1.4. Alcance

Para el presente trabajo de titulación se contempla la automatización de la planificación de la logística de última milla, es decir, la fase de programación de la asignación de mercancías a vehículos de despacho.

No obstante, únicamente se pretende automatizar la logística de última milla de cualquier empresa cuya distribución de mercancías consista en la salida de vehículos desde una sede principal hacia diversos puntos de entrega, considerando la variabilidad de la demanda del día en cuanto a los pedidos de sus clientes y sus respectivas ubicaciones; es decir, se contempla la automatización de la logística de última milla de todas las empresas cuyo planteamiento de distribución de mercancía consista en un único origen y varios destinos.

Por otra parte, no se automatizarán procesos “anormales” como la asignación de mercancías a vehículos de despacho en las que su peso o volumen reportado sea mayor que la capacidad máxima del vehículo con mayor capacidad de la flota de la empresa considerada.

CAPÍTULO II: REVISIÓN SISTEMÁTICA DE LITERATURA

2. Revisión sistemática de literatura

Para el desarrollo de la revisión sistemática de literatura, se utilizaron los siguientes buscadores:

- Ebsco
- Scielo
- Dialnet
- Springer
- Digitalia
- Jstor
- Scopus
- ScienceDirect
- Biblioteca digital de la Pontificia Universidad Católica del Ecuador
- Google Académico
- ResearchGate
- Google

A partir de los buscadores mencionados, para descubrir y seleccionar los recursos para el desarrollo de la revisión sistemática de literatura, se tomaron en cuenta las siguientes palabras clave y sus combinaciones:

- Logística
- Entrega de productos
- Logística de última milla
- Optimización de la última milla
- Machine Learning
- Aprendizaje automático
- Last-Mile Optimization
- VRP (Vehicle Routing Problem)

A partir de la búsqueda de las palabras clave catalogadas en los buscadores, se identificaron 20 recursos bibliográficos entre artículos científicos y páginas web, sin embargo, un porcentaje de estas fuentes fueron descartadas tras una evaluación mediante los siguientes criterios de selección:

- Para páginas web, se seleccionaron fuentes que permitan explicar de manera adecuada los conceptos requeridos para comprender de manera preliminar en que consiste la logística de última milla y en cómo podría optimizarse a través de la aplicación de técnicas de inteligencia artificial.
- Para artículos científicos se seleccionaron aquellos que permitan tener una recopilación de lo que se ha estado trabajando durante los últimos años con relación a la optimización de la última milla, además, se seleccionaron aquellos

que puedan brindar ideas y un acercamiento a como se debe planificar la solución del problema planteado en la presente tesis.

Por lo tanto, a partir de la evaluación de los recursos bibliográficos mediante los criterios de selección descritos, se determinó el uso de 12 fuentes para el desarrollo de la revisión sistemática de la literatura, en donde, a partir de estas fuentes se ha planificado el presente capítulo mediante tres ejes capaces de sustentar las bases con las que se realizará esta tesis, estas son:

- **Logística de la Última Milla:** se realizará una revisión de lo que consiste la logística de última milla, su importancia, problemáticas, criterios de evaluación y estructura.
- **Inteligencia Artificial y la optimización de la logística de la última milla:** se expondrá como, a partir del avance del tiempo y de la aparición de la cuarta revolución industrial, la inteligencia artificial se ha posicionado como un pilar fundamental para la optimización de la logística de la última milla.
- **El Vehicle Routing Problem (VRP) y la logística de la Última Milla:** se explicará qué es el Vehicle Routing Problem, cómo está relacionada con la logística de la última milla y cuáles son los enfoques más recientes para su resolución.

2.1. Logística de la Última Milla

La logística de última milla ha logrado adquirir una variedad de definiciones en las que todas hacen alusión al último tramo de la cadena de suministro, la cual consiste en “el movimiento de mercancías desde un centro de transporte hasta su destino final” (Bopage,

Vidanagamachchi, & Julian, 2019, pág. 2), es decir, la logística de última milla es “el último tramo de una cadena de suministro de una empresa que parte desde el punto de generación de un pedido hasta el punto de destino final preferido por el cliente para su recepción” (Lim, Xin Jin, & Singh Srai, 2015, pág. 2).

Adicionalmente, la logística de última milla es considerada “uno de los segmentos más complicados, caros e ineficientes a lo largo de toda la cadena de suministro” (Gevaers, Van de Voorde, & Vanelslander, 2014, pág. 1), principalmente debido a que en este segmento de la cadena se encuentra una multitud de problemas que afecta directamente a la entrega de mercancías, produciendo así una secuela irremediable en los costos y la satisfacción del cliente.

En este sentido, se prevé la importancia de identificar los problemas principales a los que se ve afectada la logística de última milla, los cuales son “las entregas tardías, la ausencia del cliente en su lugar de negocio, altos costes de entrega y largo tiempo de entrega” (Ousaid, 2020, pág. 27), esto a su vez, es sustentado por Gevaers, Van de Voorde y Vanelslander al mencionar que cuando los pedidos solicitados requieren de firma y el cliente no se encuentra en su lugar de negocio, entonces, será necesario enviar nuevamente el paquete para que se entregue con éxito (Gevaers, Van de Voorde, & Vanelslander, 2014, pág. 2). Por lo que, habiéndose identificado que estas entregas que deben realizarse por segunda vez representan el 12% del total, y que, aquellas que ya no son entregables son alrededor del 2%, es inevitable que generen un coste adicional. (Ousaid, 2020, pág. 27).

Por tal motivo, es importante tener en cuenta ciertos aspectos clave adecuados para evaluar las soluciones logísticas de última milla, pues, “las capacidades logísticas pueden hacer contribuciones importantes tanto en la estrategia como en el desempeño corporativo, e incluso, en proporcionar una base competitiva para crear un valor diferenciado para el cliente” (Morash,

Droge, & Vickery, 1996, pág. 1), en otras palabras, la logística de última milla tiene la capacidad de hacer que una empresa tenga un valor adicional con la cual pueda ganarse la lealtad de actuales y nuevos clientes.

No obstante, para lograr esto es necesario comprender la manera en la que se dividen las capacidades logísticas mencionadas, estas son: capacidades logísticas orientadas a la demanda y capacidades logísticas orientadas a la oferta; que, según Morash, “la primera disciplina de valor hace hincapié en el cliente (...) y abarca la diferenciación de productos o servicios y mejoras a partir de capacidades logísticas como ventajas de tiempo” (Morash, Droge, & Vickery, 1996, pág. 2), mientras que, las capacidades de logística orientadas a la oferta “están relacionadas con las capacidades operativas de una organización” (Morash, Droge, & Vickery, 1996, pág. 2).

Con este contexto, a partir del estudio de Morash, se logró determinar que todas las capacidades de logística orientadas tanto a la demanda como a la oferta son relevantes para el rendimiento adecuado de una empresa (Morash, Droge, & Vickery, 1996, pág. 15), sin embargo, a partir de las correlaciones entre las capacidades logísticas estratégicas y el desempeño de la empresa, se identificó cuatro capacidades logísticas clave, estas son: velocidad de entrega, confiabilidad, capacidad de respuesta y distribución de bajo costo; las cuales representan las mejores opciones para generar una ventaja competitiva sostenida (Morash, Droge, & Vickery, 1996, pág. 10), y por tanto, pueden ser identificados como los pilares adecuados para evaluar una solución logística de última milla.

Finalmente, es vital comprender que la logística de última milla consiste en un sistema de movimiento lineal de mercancías, en donde, existen tres tipos de estructuras para comprender su funcionamiento, las cuales son: “(1) push centric: mercancía 'enviada' a los clientes; (2) pull centric: mercancía 'obtenida' de la fuente de la mercancía por el cliente; (3) hybrid: mercancía

'enviada' a algún sitio intermedio donde la mercancía será 'obtenida' por el cliente" (Lim, Xin Jin, & Singh Srai, 2015, pág. 4).

2.2. Inteligencia Artificial y la optimización de la logística de la Última Milla

Como todas las industrias, la industria de la logística continuamente se enfrenta a una gran cantidad de desafíos y problemas que, además de obligar a desarrollar innovaciones, también implica el resolver las cosas de manera diferente (Jucha, 2021, pág. 2), de manera que, sumado a todos los avances recientes guiados principalmente por la cuarta revolución industrial, nace la necesidad de incorporar nuevas tecnologías dentro de los procesos de logística.

De hecho, "la adopción de tecnologías 4.0 puede jugar un papel crucial para intentar optimizar el servicio, en términos de digitalización, automatización e interconexión no solo a lo largo de la cadena de suministro, sino también en su última milla" (Ferrari A., Mangano G., Carlin A., 2021, como se citó en López & Flórez, 2023, pág. 2).

Por lo que, resulta de vital importancia ya que la necesidad de innovación en la logística de última milla ha crecido exponencialmente a lo largo de los últimos años, principalmente debido a la pandemia de la Covid 19 que impulsó el comercio electrónico y por consiguiente las entregas en los sitios de preferencia de los clientes (López & Flórez, 2023, pág. 10), lo que se evidencia porque "empresas como Amazon y Alibaba están invirtiendo cientos de millones de dólares para hacer que sus procesos de entrega sean mucho más rápidos y eficientes para garantizar la entrega al día siguiente o incluso el mismo día" (Open Data Science, 2021).

Es más, la inteligencia artificial está adquiriendo cada vez más relevancia en una gran variedad de industrias, sin embargo, particularmente en la logística, su importancia ha crecido principalmente debido a "las crecientes demandas de los clientes y el crecimiento de los

volúmenes de envío que están generando dificultades (...) especialmente en la última milla” (Rosendorff, Hodes, & Fabian, 2021, pág. 1), en este sentido, las nuevas tecnologías potenciadas con inteligencia artificial se han destacado por “ser muy útiles en la entrega de última milla puesto que agilizan la entrega de pedidos a los clientes y ahorran costes optimizando todo el proceso de entrega” (Jucha, 2021, pág. 7).

En otras palabras ,la inteligencia artificial durante los últimos años ha demostrado ser una herramienta valiosa que ha presentado un gran potencial para la optimización de la logística de última milla.

2.3. El Vehicle Routing Problem (VRP) y la logística de la Última Milla

Para empezar, el problema de ruteo de vehículos o VRP por sus siglas en inglés, es considerado “un problema complejo de gestión logística y representa una clase clave de problemas a resolver para reducir costos y optimizar los recursos logísticos” (Open Data Science, 2021), más aún, estos VRP “son modelos normalmente destinados para apoyar la toma de decisiones relativas al ruteo de vehículos comerciales o de servicios, típicamente camiones o medios de transporte similares” (Mor & Speranza, 2020, pág. 2).

Donde, su objetivo principal es “calcular una ruta en condiciones específicas que minimice el coste total del transporte, al reducir, por ejemplo, la distancia total recorrida, el número de vehículos utilizados o el tiempo total de transporte” (Jucha, 2021, pág. 5), a través de decisiones que “refieren a la asignación de clientes a los vehículos y a la secuenciación de los clientes asignados a cada vehículo de tal manera que se minimice el coste total de la ruta” (Mor & Speranza, 2020, pág. 2).

Por otra parte, cuando se prevé la resolución de un VRP en la logística de última milla de un problema de la vida real, es inevitable evitar que surjan una serie de problemas que pueden “diferir debido a características adicionales que debe satisfacer una ruta como una restricción en la duración, el respeto de las ventanas horarias o las secuencias de operaciones de recogida y entrega” (Mor & Speranza, 2020, pág. 2).

Además, al resolver un VRP en el mundo real, “existen varios objetivos o medidas de desempeño que a menudo son contradictorios, pues, en algunos sectores la satisfacción del cliente y la entrega oportuna son más importantes que minimizar la distancia recorrida” (Giuffrida, y otros, 2022, pág. 8), por lo que, es de vital importancia considerar estos distintos factores, condiciones y restricciones para que la resolución de un VRP pueda proveer beneficios como “el ahorro de tiempo, distancia y combustible y la satisfacción del cliente gracias a tiempos de entrega más cortos” (Open Data Science, 2021).

De hecho, “desde un punto de vista matemático y computacional, no existe un único tipo de problema de ruteo de vehículos sino varios” (Hanne, 2009, pág. 2), es decir, con base a los distintos factores, condiciones y restricciones mencionadas, se pueden generar distintas subvariantes de un VRP en el que la asignación de clientes a vehículos y rutas pueda resultar en distintos niveles de complejidad, sin embargo, de todas maneras, “el clásico VRP y todas las subvariantes son NP-difíciles, es decir, no existe un determinado algoritmo que asegure el hallazgo de una solución óptima” (Giuffrida, y otros, 2022, pág. 8).

De esta forma, aunque ya existen varios métodos que pueden obtener buenos resultados generales en algunos casos de problemas de generación de rutas para vehículos, ninguno de ellos funciona bien para todas las situaciones (Gutierrez Rodríguez, Conant Pablos, Ortiz Bayliss, & Terashima Marín, 2019, pág. 1).

Es por ello por lo que se han propuesto nuevos enfoques para solventar los VRPs y sus posibles variantes a través de métodos híbridos donde se aplican distintos algoritmos y procesos para distintas etapas de la resolución de un VRP, entre los que, destacan la aplicación de modelos de aprendizaje automático (Gutierrez Rodríguez, Conant Pablos, Ortíz Bayliss, & Terashima Marín, 2019, pág. 2), con los que se han logrado obtener resultados favorables (Giuffrida, y otros, 2022, pág. 8).

En otras palabras, la aplicación y uso de modelos de aprendizaje automático ha demostrado optimizar la resolución de VRPs y, por ende, la gestión de la logística de última milla.

CAPÍTULO III: MARCO TEÓRICO

3. Marco Teórico

3.1. CRISP-DM

CRISP-DM es un modelo ampliamente adoptado para estructurar proyectos de minería de datos pues su flexibilidad y adaptabilidad lo han convertido en un estándar de facto en la industria (IBM, 2021).

Este modelo comprende seis fases interdependientes que permiten la ejecución eficiente y eficaz de un proyecto de minería de datos, sin embargo, es necesario notar que CRISP-DM no impone una secuencia rígida entre sus fases, de hecho, es común que los proyectos se muevan de ida y vuelta entre las diferentes fases según sea necesario, ya que, la salida de cada una determina cuál será la siguiente fase o tarea por realizar (Chapman, y otros, 1999).

En la imagen a continuación se podrán observar las fases con las que consta CRISP-DM, no obstante, de igual manera se ilustran flechas que indican las dependencias más importantes y frecuentes entre estas, además, también se puede visualizar un círculo exterior en el diagrama el cual simboliza la naturaleza cíclica del proceso, pues, este no termina una vez que se despliega una solución, ya que, “las lecciones aprendidas durante el proceso y de la solución implementada pueden desencadenar nuevas preguntas de negocio, a menudo más enfocadas y específicas, que a su vez inician nuevos ciclos de minería de datos” (Chapman, y otros, 1999).

Ilustración 1

Ciclo de vida de la minería de datos



Nota. Ciclo de vida de la minería de datos, por IBM, 2021, IBM (<https://www.ibm.com/docs/en/spss-modeler/saas?topic=dm-crisp-help-overview>).

3.1.1. Entendimiento del negocio

Esta primera fase del modelo CRISP-DM es crucial ya que establece la base para todo el proyecto de minería de datos que se esté planteando, pues, “su objetivo principal es comprender los objetivos y requisitos del proyecto desde una perspectiva del negocio y convertir este conocimiento en una definición como problema de minería de datos junto con un plan preliminar para lograr dichos objetivos” (Chapman, y otros, 1999).

Es decir, esta fase se compone de las siguientes tareas:

1. Determinación de los objetivos del negocio
2. Evaluación de la situación

3. Determinación de los objetivos de minería de datos
4. Producción de un plan de proyecto

Con este contexto, para empezar un proyecto de minería de datos, el primer objetivo es comprender a fondo, desde una perspectiva de negocio, lo que el cliente o el propio negocio realmente desea lograr; posteriormente, es necesario realizar una evaluación de la situación que consiste principalmente en una “investigación detallada sobre todos los recursos, restricciones, supuestos y otros factores que deben considerarse para determinar los objetivos del análisis de datos y el plan del proyecto” (Chapman, y otros, 1999).

Una vez finalizada la evaluación de la situación y teniendo claro el objetivo del negocio, se debe proceder con su traducción en una realidad de minería de datos, es decir, en un objetivo de lo que se plantea realizar mediante la aplicación de métodos de minería de datos como, por ejemplo y para esta tesis, aprendizaje automático.

Finalmente, se debe construir un plan del proyecto preliminar donde se describa cómo se lograrán los objetivos de minería de datos y, por ende, los de negocio; en otras palabras, en el plan del proyecto debe especificar los pasos a seguir durante el resto del proyecto (Chapman, y otros, 1999).

3.1.2. Entendimiento de los datos

Esta es la segunda fase del modelo CRISP-DM y comienza con una recopilación inicial de datos y tiene el objetivo de familiarizarse con ellos, identificar problemas de calidad y realizar descubrimientos preliminares sobre ellos (Chapman, y otros, 1999); en otras palabras, esta fase busca obtener una comprensión profunda de los datos disponibles para la minería con el fin de evitar problemas inesperados durante la fase de preparación de los datos (IBM, 2021).

3.1.3. Preparación de los datos

La fase de preparación de los datos es la tercera etapa del modelo y abarca todas las actividades necesarias para construir un conjunto de datos final que será utilizado para la elaboración de modelos, así como, para distintas otras aplicaciones.

La preparación de los datos “es uno de los aspectos más importantes y a menudo más laboriosos de la minería de datos, pues, se estima que esta fase consume entre el 50% y el 70% del tiempo y esfuerzo de un proyecto” (IBM, 2021), no obstante, cabe mencionar que, si se distribuye suficiente dedicación a las fases anteriores de entendimiento del negocio y de los datos, se puede minimizar el esfuerzo requerido para esta.

Entre las actividades más comunes de esta fase se pueden encontrar la siguientes:

- Selección de variables relevantes.
- Limpieza de datos.
- Construcción de nuevas variables.
- Integración de conjuntos de datos.
- Formateo de datos y variables.

3.1.4. Modelado

Esta es la cuarta fase del modelo CRISP-DM y es en la que se “seleccionan y aplican diversas técnicas de modelado, y se calibran sus parámetros para obtener resultados óptimos.” (Chapman, y otros, 1999), en otras palabras, el objetivo principal de la fase de modelado es desarrollar uno o más modelos que puedan predecir o clasificar con precisión los resultados

esperados basándose en los datos preparados (IBM, 2021), resultados que están estrechamente ligados a los objetivos del negocio definidos en la primera fase.

Entre las principales tareas que conlleva esta fase están:

- Selección de las técnicas de modelado.
- Construcción de los modelos.
- Evaluación de los modelos.

3.1.5. Evaluación

Esta fase surge una vez se han construido “uno o más modelos que aparentan poseer una calidad alta desde una perspectiva de análisis de datos, por lo que, antes de proceder con su implementación final, es fundamental asegurarse que dichos modelos se acoplan adecuadamente con los objetivos del negocio” (Chapman, y otros, 1999), con este contexto, el objetivo principal de esta fase es determinar si el modelo cumple con los criterios de éxito de negocio establecidos durante la primera fase, por lo que, es esencial verificar que no haya ninguna cuestión del negocio importante que no se haya considerado suficientemente (IBM, 2021).

3.1.6. Despliegue

Esta es la última fase del modelo CRISP-DM y su objetivo principal es utilizar los nuevos conocimientos obtenidos para realizar mejoras dentro de la organización (IBM, 2021), sin embargo, dependiendo de los requisitos, “la fase de despliegue puede ser tan simple como generar un informe o tan compleja como implementar un proceso de minería de datos repetible en toda la empresa” (Chapman, y otros, 1999).

3.2. Aprendizaje automático no supervisado

El aprendizaje automático no supervisado “es un tipo de aprendizaje automático en el que los modelos aprenden a partir de conjuntos de datos sin etiquetar sobre el que se les permite actuar sin supervisión” (INESDI, 2022).

Una de las principales diferencias entre el aprendizaje automático supervisado y no supervisado “es la necesidad de datos etiquetados, ya que, mientras que el aprendizaje supervisado se basa en datos de entrada y salida etiquetados, el aprendizaje no supervisado procesa datos sin etiquetar o sin procesar” (Universidad Europea, 2022).

En otras palabras, el aprendizaje automático no supervisado es una técnica de inteligencia artificial que parte de la rama de aprendizaje automático que funciona a partir de datos en los que desconocemos como se pueden relacionar de acuerdo con ciertas características.

3.3. Agrupación en clústeres

En términos simples, “la agrupación en clústeres es una técnica de minería de datos que agrupa datos no etiquetados en función de sus similitudes o diferencias” (IBM, s.f.), este tipo de técnica “encuentra los puntos en común de los objetos que componen el conjunto de datos y los clasifica según la presencia o ausencia de estos” (INESDI, 2022).

Es decir, los algoritmos de agrupación en clústeres procesan datos para que, de acuerdo con sus características, se puedan identificar grupos representados por estructuras o patrones en la información.

CAPÍTULO IV: ENTENDIMIENTO DEL NEGOCIO

4. Entendimiento del negocio

4.1. Objetivos del negocio

Este capítulo tiene el objetivo de comprender, en profundidad, qué es lo que se desea conseguir a partir del presente trabajo de titulación, de manera que, para conseguirlo, este subcapítulo presentará una revisión de antecedentes, una comprensión del objetivo desde la perspectiva del negocio y los criterios con los que se evaluará el éxito del desarrollo del trabajo.

4.1.1. *Antecedentes*

Actualmente a Biologicsoft, una empresa de desarrollo de software que posee un sistema de planificación de logística de última milla, se le ha solicitado el desarrollar un mecanismo por el cual esta planificación se realice de manera automática, pues, la complejidad de la logística manual referente a la distribución de mercancía a distintos puntos de entrega ha causado problemas como pedidos rechazados, no entregados e ineficiencia en la planificación de rutas con lo que respecta a la asignación de documentos (véase la terminología en el capítulo 5.2.4) a vehículos para su despacho.

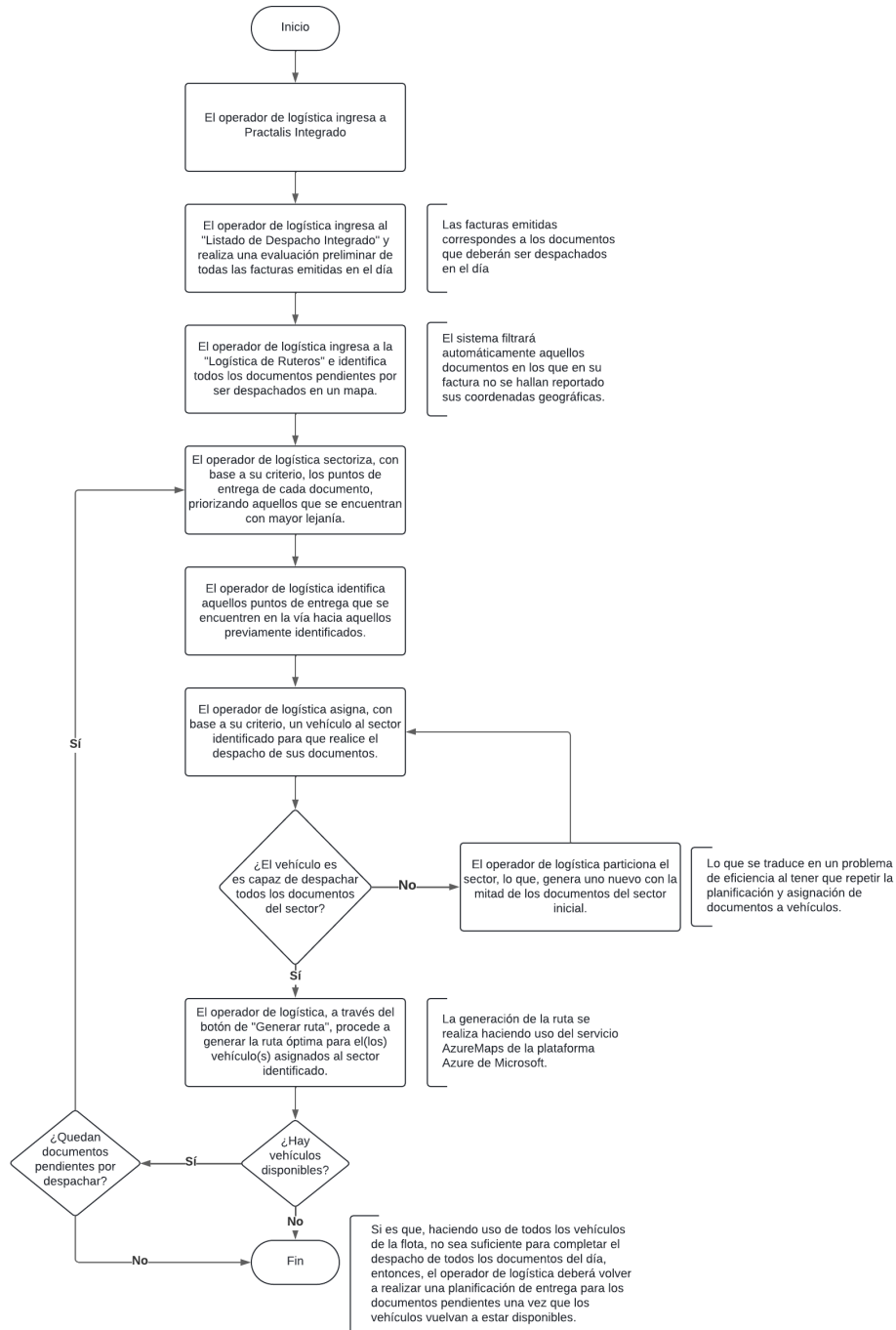
Sin embargo, además de perjudicar en los ingresos de las empresas que realizan esta planificación de manera manual, esto también ha ocasionado que distintos empleados que han sido encargados de la logística de última milla de dichas empresas deserten sus puestos por temas relacionados al estrés ocasionado por la complejidad que conlleva la planificación de esta logística.

Por esta razón, existe la necesidad de automatizar el proceso de logística de última milla llevado actualmente a cabo, por lo que, en primera instancia se ha sistematizado el proceso que

se realiza para su planificación, el cual, puede ser analizado en el diagrama de flujo a continuación:

Ilustración 2

Diagrama de flujo del proceso de logística de última milla actual



4.1.2. Objetivo del Negocio

En este sentido, el objetivo del negocio es mejorar la planificación de la logística de última milla mediante una mejor asignación de documentos a vehículos de despacho.

4.1.3. Criterios de Éxito

Desde la perspectiva del negocio, la automatización de la logística de última milla tendrá éxito si se consigue lo siguiente:

- Reducir el porcentaje de subutilización de los vehículos en un 40%.

4.2. Evaluación de la situación

Una vez comprendida de manera preliminar lo que se desea conseguir en términos del negocio, es crucial realizar una evaluación de la situación en la que se detalle el inventario de recursos disponibles para lograr cumplir el objetivo propuesto, especificar requerimientos, supuestos, limitaciones, y, finalmente, elaborar un listado de la terminología que será utilizada durante el desarrollo del trabajo.

4.2.1. Inventario de Recursos

Entre los recursos disponibles para el desarrollo de la presente tesis se encuentran los siguientes:

Tabla 1
Inventario de recursos

| Tipo de recurso | Recurso | Detalle |
|------------------------|-------------------------------|---|
| Personal | Ingeniero de software y datos | Encargado del desarrollo del sistema de Practalis y del diseño de las estructuras de los datos que se plantean utilizar para el desarrollo de la logística de última milla automatizada, es decir, es a quien se le solicitarán los datos requeridos. |

| | | |
|----------------------------------|---|---|
| Datos | Datos históricos de documentos por ser despachados | Datos históricos de documentos que han sido despachados. |
| Datos | Registro de los vehículos de la flota de la empresa a la cual corresponden los datos históricos de los documentos por ser despachados | Información detallada de todos los vehículos dedicados al despacho de mercancías de la flota de la empresa a la cual corresponden los datos históricos de los documentos por ser despachados. |
| Recursos tecnológicos y software | Jupyter Notebook | Se hará uso de Jupyter Notebook para el desarrollo de la logística de última milla automatizada. |

4.2.2. Requerimientos, Supuestos y Limitaciones

Para empezar, los principales requerimientos ligados con el desarrollo propuesto tienen como base la disponibilidad del acceso a los datos, pues, un pilar fundamental de este requerimiento es que los datos no deben estar erróneos desde la fuente, es decir, los datos relacionados a facturas que han sido emitidas y recopiladas deben ser correctas dentro de un sentido natural o real del negocio.

En otras palabras, el requerimiento no parte de abstenerse de la limpieza de datos, si no que, como menciona el antiguo refrán “Garbage in Garbage out”, es importante tener en cuenta que los datos, a pesar de que algunos puedan o no tener errores superficiales, sean fieles a la realidad y que, entre los valores nulos que posiblemente posean, estos no sean en su mayoría aquellos con relación a necesidades intrínsecas para el desarrollo de la logística de última milla automatizada, como, por ejemplo, el peso y volumen de los documentos por ser despachados, o, por su parte, de las capacidades de los vehículos que tendrán la responsabilidad de concretar sus despachos.

Con base a los requerimientos, se asume que los datos tendrán información relevante con relación tanto con las capacidades inherentes de los vehículos de transporte de mercancía en cuanto a peso y volumen, como, con información relevante de los documentos por ser despachados, es decir, que estos datos contengan variables donde se detallen sus pesos, volúmenes y ubicaciones de destino.

Finalmente, como restricción, se debe completar el desarrollo de la logística de última milla automatizada hasta antes de septiembre del 2024 tanto por las propias limitaciones referentes a la entrega y defensa del trabajo de titulación, así como el plazo máximo propuesto por Biologicsoft para la entrega del sistema.

4.2.3. Terminología

A partir de este subcapítulo se utilizará terminología específica relacionada con la temática del presente trabajo de titulación, esto, con el fin de estandarizar la comunicación y facilitar la comprensión del contenido de los siguientes capítulos y subcapítulos que forman parte del desarrollo de una logística de última milla automatizada.

- **LUMA:** Para fines prácticos y con el objetivo de ensamblar un nombre al desarrollo de una logística de última milla automatizada, a partir de este momento se definirá y llamará a dicho desarrollo como **LUMA (Logística de Última Milla Automatizada)**.
- **Documentos:** Un documento se refiere a la mercancía que ha sido solicitada por un cliente y que puede corresponder a uno o varios productos que deben ser despachados, es decir, un documento puede estar constituido por distintas

mercancías con diferentes pesos y volúmenes, pero que son sumados para obtener su total como el peso del documento.

- **VRP:** Son las siglas del vehicle routing problem o, en español, del problema de ruteo de vehículos, y representa un nombre genérico para una gran variedad de problemas relacionados a la logística de última milla.
- **Dataframe:** Un dataframe es una estructura de datos construida a base de filas y columnas, similar a una hoja Excel o una base de datos, y es el tipo de dato con el que se manejarán los conjuntos de datos que se adquirirán en futuros capítulos.

4.3. Objetivos de minería de datos

Este capítulo tiene el objetivo de traducir el objetivo del negocio a una realidad de minería de datos, es decir, se interpretará el objetivo de negocio planteado en una terminología que permita conocer con exactitud que se plantea realizar en términos de minería de datos.

4.3.1. *Objetivos de minería de datos*

- Utilizar datos históricos de documentos por ser despachados junto con su información respecto a geolocalización, peso y volumen, para construir un modelo de aprendizaje no supervisado que permita identificar sectores de puntos de entrega óptimos a partir de como estos documentos se relacionan de acuerdo con su información, sin importar la variabilidad de documentos por ser despachados por día.
- Desarrollar un algoritmo que asigne de manera óptima vehículos de despacho disponibles a los distintos sectores identificados mediante el modelo planteado.

4.4. Plan del proyecto

4.4.1. Interpretación del problema de ruteo de vehículos (VRP)

El VRP que se plantea establece que existe un determinado número de vehículos con capacidades heterogéneas (vehículos con distintas capacidades en cuanto a peso y volumen) que tienen que despachar un determinado número de documentos por día, los cuales contienen diferentes pesos y volúmenes, y que se dirigen a diferentes ubicaciones con excepción de aquellas que se emitieron por un mismo cliente.

Además, se conoce que los vehículos, por lo general, no serán capaces de completar el despacho de todos los documentos en una sola ruta, lo que quiere decir que algunos vehículos tendrán que ser reutilizados para completar el despacho de todos los documentos que han sido emitidos en el día.

4.4.2. Propuesta de solución del problema y de desarrollo de LUMA

Con la finalidad de desarrollar LUMA y que sea capaz de abarcar las condiciones especificadas durante la interpretación del VRP presente, se ha propuesto una solución basada en aprendizaje automático no supervisado mediante la aplicación de técnicas de agrupación en clústeres, esto, con la finalidad de agrupar los documentos que sean emitidos para su despacho, de acuerdo con sus características como, por ejemplo, sus coordenadas geográficas.

De esta manera, a cada grupo identificado se le considerará como un sector de entrega al cual se le asignarán uno o más vehículos dependiendo de las circunstancias que se presenten, donde, una vez realizada dicha asignación, se proceda a enviarlos a Practalis Integrado para que realice la generación de una ruta óptima para cada vehículo haciendo uso del servicio Azure Maps.

Finalmente, una vez conseguidas las distintas rutas por vehículo, se reportará al sistema Practalis que vehículos están despachando documentos, el estado de los documentos en cuanto hayan sido o no asignados a algún vehículo y, por último, las rutas óptimas por vehículo.

En este sentido, lo que se busca a través del uso del agrupamiento en clústeres es identificar sectores de entrega, es decir, sectores donde se ubiquen los puntos de entrega de los distintos documentos con la finalidad de asignar a cada uno de estos sectores un vehículo que sea capaz de despachar todos, o al menos la gran mayoría, de los documentos del sector en un solo viaje.

Por lo que, el VRP interpretado anteriormente se transformará a tener un único vehículo y un determinado número de documentos que el vehículo podrá despachar, que, en consecuencia, provocará que el VRP resultante únicamente conlleve el cálculo de la ruta óptima para la entrega de todos los documentos. Dicho cálculo será realizado por Practalis Integrado mediante el uso del servicio AzureMaps.

No obstante, con respecto a aquellos VRPs en los que sus vehículos no sean capaces de despachar todos sus documentos, entonces, el vehículo deberá realizar el despacho de los documentos que le sean posibles; para que, una vez que al menos un vehículo finalice su ruta, el operador de logística pueda volver a generar una nueva planificación automática con los vehículos de despacho disponibles hasta finalizar la entrega de los documentos restantes.

CAPÍTULO V: ENTENDIMIENTO DE LOS DATOS

5. Entendimiento de los datos

Durante esta fase se plantea la recopilación, descripción, exploración y verificación de la calidad de los datos que serán necesarios para el desarrollo del presente trabajo, es decir, para el desarrollo de LUMA.

5.1. Recopilación de datos

En esta primera etapa el objetivo es recopilar los datos necesarios para el desarrollo de LUMA, para esto, la organización del subcapítulo conlleva una sección en la que se describen los datos que se requieren y cuáles fueron adquiridos, así como también, un resumen de los datos recolectados.

5.1.1. Descripción de los datos requeridos

Para el desarrollo de LUMA en el que se plantea el uso de aprendizaje automático no supervisado para la sectorización de los documentos por ser despachados para que, posteriormente, se asignen vehículos a estos sectores, resulta natural que los datos que se requieren sean los siguientes:

- Datos históricos con información de documentos que han sido despachados a través del tiempo, ya que, permitirá realizar un modelo que, independientemente de cuáles y cuántos documentos sean emitidos para su despacho en un día dado, pueda ser capaz de sectorizarlos y posteriormente asignar vehículos a dichos sectores.

- Datos de los vehículos de la flota de la empresa a la cual pertenecen los datos de los documentos que han sido despachados, debido a que, con esta información, se será capaz de asignar estos vehículos a los distintos sectores identificados.

5.1.2. Resumen de los datos recolectados

En resumen, se recolectaron dos conjuntos de datos que cumplen con las expectativas descritas en el anterior subcapítulo, estos son:

Tabla 2
Resumen de los datos recolectados

| Nombre del conjunto de datos | Formato | Método de adquisición | Detalle |
|-------------------------------------|------------------|---|---|
| Documentos | Archivo de Excel | Solicitud al ingeniero de software y datos de Biologicsoft, identificado dentro de los recursos del proyecto. | Contiene información detallada e histórica de documentos que han sido emitidos y despachados a partir del 11 de marzo del 2024 hasta el 11 de abril del 2024. |
| Vehículos | Archivo de Excel | Solicitud al ingeniero de software y datos de Biologicsoft, identificado dentro de los recursos del proyecto. | Contiene información detallada de los vehículos disponibles de la flota de la empresa a la cual pertenecen los datos de los documentos que se han despachado. |

5.2. Descripción de los datos

Una vez recolectado los conjuntos datos descritos en el subcapítulo anterior, es necesario comprenderlos, por lo que, este subcapítulo tiene el objetivo de describir a detalle estos conjuntos en cuanto a sus dimensiones y variables disponibles; además, también tiene la finalidad de evaluar preliminarmente si estos datos satisfacen los requerimientos necesarios para lograr un adecuado desarrollo de LUMA.

5.2.1. Descripción del conjunto de datos de Documentos

Para comenzar, se procederá con la carga de los datos, para lo cual, se utilizará la librería pandas junto con la función `read_excel`:

Ilustración 3

Carga del conjunto de datos de Documentos

```
documentos = pd.read_excel(ubicacion + "Documentos.xlsx")
```

Una vez cargado los datos referentes a los documentos que han sido emitidos y despachados a partir del 11 de marzo hasta el 11 de abril, se procederá a evaluar sus dimensiones:

Ilustración 4

Dimensiones del conjunto de datos de Documentos

Dimensiones de los datos

```
print(f"Los datos contienen {documentos.shape[0]} registros y {documentos.shape[1]} variables.")
```

```
Los datos contienen 1602 registros y 45 variables.
```

Como es visible, el conjunto de datos dispone de 1602 registros de documentos despachados y de 45 variables, las cuales serán descritas a continuación con la finalidad de evaluar si los datos adquiridos contienen información relevante.

No obstante, es importante notar que se ha dividido la descripción de las distintas variables según su naturaleza con el propósito de proveer de una mejor comprensión del sentido y descripción de estas.

Por lo que, para empezar, las tablas de la 3 a la 7 corresponden a las descripciones de las variables que tienen el objetivo de identificar a los distintos documentos, clientes, despachadores y vehículos; estas son:

Tabla 3

Variables utilizadas para la identificación de los documentos

| Variable | Descripción |
|----------------------|---|
| codigoEmpresa | Es una extensión del código único utilizado para la identificación de un documento. |
| codigoFacilidad | Es una extensión del código único utilizado para la identificación de un documento. |
| codigoCaja | Es una extensión del código único utilizado para la identificación de un documento. |
| codigoTipoDocumento | Es una extensión del código único utilizado para la identificación de un documento. |
| codigoSerieDocumento | Es una extensión del código único utilizado para la identificación de un documento. |
| codigoDocumento | Es una extensión del código único utilizado para la identificación de un documento. |

Tabla 4

Variables utilizadas para la identificación de los despachadores

| Variable | Descripción |
|------------------------|---|
| codigoEmpresa | Es una extensión del código único utilizado para la identificación de un despachador. |
| codigoDespachador | Es una extensión del código único utilizado para la identificación de un despachador. |
| sucursalDespachador | Es una extensión del código único utilizado para la identificación de un despachador. |
| informacionDespachador | Corresponde al nombre de un despachador. |

Tabla 5

Variables utilizadas para la identificación de los vehículos

| Variable | Descripción |
|-----------------|--|
| codigoEmpresa | Es una extensión del código único utilizado para la identificación de un vehículo. |
| codigoFacilidad | Es una extensión del código único utilizado para la identificación de un vehículo. |
| codigoVehiculo | Es una extensión del código único utilizado para la identificación de un vehículo. |

informacionVehiculo Corresponde a la información de la marca, modelo y placa del vehículo.

Tabla 6
Variables utilizadas para la identificación de los clientes

| Variable | Descripción |
|-----------------|---|
| codigoEmpresa | Es una extensión del código único utilizado para la identificación de un cliente. |
| codigoTercero | Es una extensión del código único utilizado para la identificación de un cliente. |
| sucursalTercero | Es una extensión del código único utilizado para la identificación de un cliente. |

Tabla 7
Variables utilizadas para la identificación de las direcciones principales de los clientes

| Variable | Descripción |
|------------------------|---|
| codigoEmpresa | Es una extensión del código único utilizado para la identificación de la dirección principal del cliente donde se realizará la entrega del documento que ha solicitado. |
| codigoTercero | Es una extensión del código único utilizado para la identificación de la dirección principal del cliente donde se realizará la entrega del documento que ha solicitado. |
| sucursalTercero | Es una extensión del código único utilizado para la identificación de la dirección principal del cliente donde se realizará la entrega del documento que ha solicitado. |
| codigoTerceroDireccion | Es una extensión del código único utilizado para la identificación de la dirección principal del cliente donde se realizará la entrega del documento que ha solicitado. |

Una vez identificadas las variables que permiten la identificación de los documentos, así como de los despachadores, vehículos y clientes; a continuación, se exponen las variables que presentan información geográfica de los puntos de entrega de los clientes, información

cronológica de los documentos e información detallada del contenido y estado de estos documentos.

Tabla 8
Variables con información geográfica de los clientes

| Variable | Descripción |
|-----------------------|--|
| coordenadaLatitud | Coordenada de la latitud de la ubicación en la que el cliente desea recibir el documento solicitado. |
| coordenadaLongitud | Coordenada de la longitud de la ubicación en la que el cliente desea recibir el documento solicitado. |
| direccionPrincipal | Dirección principal de la ubicación en la que el cliente desea recibir el documento solicitado. |
| direccionInterseccion | Intersección de la dirección de la ubicación en la que el cliente desea recibir el documento solicitado. |
| direccionNumero | El número de la ubicación en la que el cliente desea recibir el documento solicitado. |
| direccionReferencia | Referencias cercanas a la ubicación en la que el cliente desea recibir el documento solicitado. |

Tabla 9
Variables cronológica de los documentos

| Variable | Descripción |
|------------------|---|
| fechaElaboracion | Es la fecha en la que se emitió la factura, es decir, la fecha de cuando se reportó el documento por ser despachado. |
| fechaVencimiento | Es la fecha máxima de pago por el documento por parte del cliente, no obstante, es importante tener en cuenta que esta variable no representa la fecha máxima de entrega de este. |
| fechaEntrega | Es la fecha de cuando el cliente solicito que se entregue el documento, y es con esta fecha con la que se realizan las consultas para la planificación del despacho. |

Tabla 10
Variables con información de los documentos

| Variable | Descripción |
|-----------------|--------------------|
|-----------------|--------------------|

| | |
|------------------------|--|
| totalDocumento | Es el valor neto del documento, es decir, es el total de todos los valores que consideran el precio, descuentos e IVA de la mercancía que forma parte de un documento. |
| totalPesoDocumento | Es el peso total del documento. |
| totalVolumenDocumento | Es el volumen total del documento. |
| estadoDocumento | Es un campo que identifica la instancia en la cual se encuentra un documento, de este, existen varias instancias que serán descritas más adelante en una tabla particular. |
| estadoProceso | El estado proceso identifica cuando un documento está confirmado para su despacho. |
| estadoProcesoDetalle | El estado proceso indica cuando un documento está autorizado para su despacho. |
| estadoProcesoActividad | Identifica el subproceso de facturación en el que se encuentra el documento, es decir, si se generó el XML, si se firmó, si se envió por correo, etc. |

En la tabla anterior se identificó la variable *estadoDocumento*, esta variable es de vital importancia ya que tiene el objetivo de reportar el estado de un documento y, por ende, permitir identificar si este ha sido despachado o si aún es necesario planificar su entrega, por lo que, es importante conocer cuáles son todos los posibles códigos que representan a las instancias por las que un documento transcurrirá hasta su entrega al cliente, estas son:

Tabla 11
Instancias de estados de documentos de la variable *estadoDocumento*

| Código | Estado | Descripción |
|--------|------------|---|
| 3 | Facturado | Significa que el documento ha sido facturado y está listo para la planificación de su despacho, es decir, el documento original antes de este estado era un pedido y se convirtió en una factura. |
| 4 | Empacado | Representa cuando un documento ha sido asignado a un vehículo. |
| 5 | Despachado | Significa cuando el documentos se encuentra en proceso de despacho, es decir, cuando el documento se encuentra en circulación. |
| 6 | Entregado | Identifica cuando el documento se ha entregado al cliente. |

Por otro lado, con el sentido de dar fin a la descripción de las variables del conjunto de datos de documentos, a continuación, se presentarán las últimas disponibles junto con sus respectivas descripciones:

Tabla 12

Variables con información de la empresa a la que pertenecen los datos

| Variable | Descripción |
|-----------------|---|
| nombreFacilidad | Es el nombre del establecimiento de la facilidad donde saldrán los vehículos. |
| seriePuntoVenta | Es la serie del establecimiento. |
| nombreCaja | Es el nombre de la caja donde se realizó la venta. |
| serieCaja | Es la serie de la caja donde se realizó la venta. |

Tabla 13

Variables con información de apoyo

| Variable | Descripción |
|-------------------------|--|
| codigoUnificado | Es una extensión del código utilizado para la identificación del cliente en otro sistema que ayuda a mantener las relaciones entre este sistema y el sistema de Practalis Integrado. |
| codigoUnificado02 | Es una extensión del código utilizado para la identificación del cliente en otro sistema que ayuda a mantener las relaciones entre este sistema y el sistema de Practalis Integrado. |
| nombreEmpresaTercero | Es la razón social del cliente. |
| nombreComercialTercero | Es la razón comercial del cliente. |
| tipoIdentificacion | Es el tipo de identificación del cliente. |
| documentoIdentificacion | Es la cédula o RUC del cliente. |
| telefono01 | Es el teléfono primario del cliente. |
| telefono02 | Es el teléfono secundario del cliente. |
| telefono03 | Es el teléfono terciario del cliente. |
| correoElectronico | Es el correo electrónico del cliente. |

5.2.2. Descripción del conjuntos de datos de Vehículos

De la misma forma que para la descripción del anterior conjunto de datos, para la descripción del conjunto de datos de Vehículos se procederá, en primera instancia, con la carga de sus datos.

Ilustración 5

Carga del conjunto de datos de Vehículos

```
vehiculos = pd.read_excel(ubicacion + "Vehículos.xlsx")
```

Una vez cargado el conjunto de datos con la información de todos los vehículos disponibles en la flota de la empresa a la cual pertenecen la información del conjunto de datos de Documentos, es necesario proceder con la evaluación de sus dimensiones:

Ilustración 6

Dimensiones del conjunto de datos de Vehículos

```
print(f"Los datos contienen {vehiculos.shape[0]} registros y {vehiculos.shape[1]} variables.")  
Los datos contienen 9 registros y 18 variables.
```

De esta forma, las dimensiones de los datos evidencian que la empresa en cuestión posee 9 vehículos en su flota, los cuales, a su vez, poseen 18 variables que los describen, estas son:

Tabla 14

VARIABLES UTILIZADAS PARA LA IDENTIFICACIÓN DE LOS VEHÍCULOS

| Variable | Descripción |
|-----------------|--|
| codigoEmpresa | Es una extensión del código único utilizado para la identificación de un vehículo. |
| codigoFacilidad | Es una extensión del código único utilizado para la identificación de un vehículo. |
| codigoVehiculo | Es una extensión del código único utilizado para la identificación de un vehículo. |

Tabla 15

Variables con información de los vehículos

| Variable | Descripción |
|-----------------------|--|
| marcaVehiculo | Es la marca del vehículo. |
| modeloVehiculo | Corresponde al modelo del vehículo. |
| anioVehiculo | Corresponde al año de producción del vehículo. |
| numeroPlacas | Son los números de la placa del vehículo. |
| numeroMatricula | Son los números de la matrícula del vehículo. |
| ultimoAnioMatricula | Corresponde al último año en que se matriculo al vehículo. |
| numeroChasis | Es el número del chasis. |
| tieneRastreoVehicular | Identifica si el vehículo cuenta con rastreo vehicular. |
| tieneSeguro | Identifica si el vehículo tiene seguro. |
| estadoVehiculo | Identifica el estado del vehículo. |

Tabla 16

Variables con información de la capacidad de los vehículos

| Variable | Descripción |
|-------------------|--|
| capacidadVehiculo | Corresponde a la capacidad total del vehículo con respecto al peso. |
| capacidadUnidad | Es la unidad de medida de la capacidad con respecto al peso del vehículo. |
| capacidadVolumen | Corresponde a la capacidad total del vehículo con respecto al volumen. |
| volumenUnidad | Es la unidad de medida de la capacidad con respecto al volumen del vehículo. |

Tabla 17

Variables con información de apoyo

| Variable | Descripción |
|-----------------|---|
| codigoUnificado | Corresponde al código único que se utiliza para generar un enlace con otro sistema y Practalis Integrado. |

5.3. Exploración de los datos

Una vez descritas las variables de los conjuntos de datos que serán utilizados para el desarrollo de LUMA, es necesario mejorar su comprensión y permitir evaluar si, con respecto a sus datos, los conjuntos son adecuados para alcanzar los objetivos propuestos en este trabajo.

Por esta razón, en este capítulo se presentarán los hallazgos que se identificaron durante la exploración de datos dividido en tres secciones:

- Exploración de datos de Documentos
- Exploración de datos de Vehículos
- Exploración de los datos de Documentos y Vehículos integrados

En las que, en la primera sección se reportarán todos los hallazgos relevantes identificados en el conjunto de datos de Documentos, mientras que, en la segunda sección se detallaran los hallazgos del conjunto de datos de Vehículos, y, finalmente, en la última sección se reportará los hallazgos de la exploración de datos de un dataframe en el que los datos de Documentos y Vehículos hayan sido integrados.

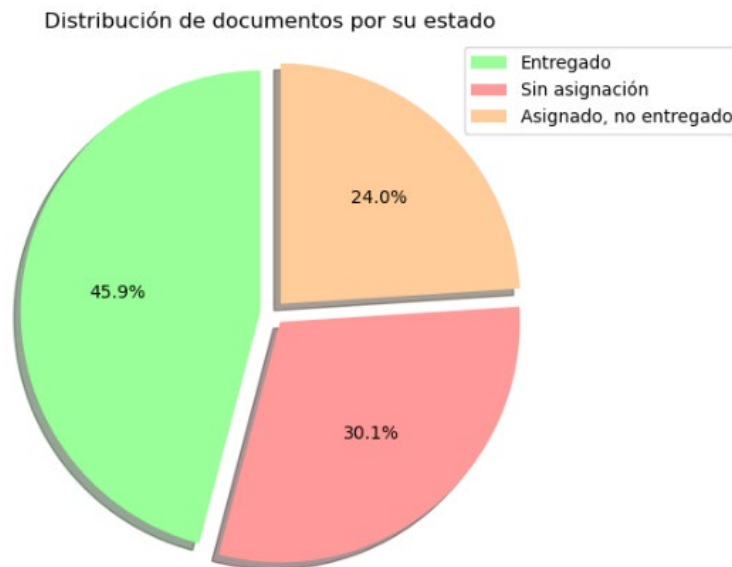
5.3.1. Hallazgos de la exploración de los datos de Documentos

Para empezar, se corroboró que los registros de documentos despachados corresponden a las fechas dentro del rango del 11 de marzo hasta el 11 de abril del 2024, donde se observó que se registraron documentos todos los días a excepción de los fines de semana y el feriado del 29 de abril por viernes santo, lo que resulta en que los datos registrados correspondan únicamente a 23 días del rango de fechas mencionado.

A su vez, se identificó que la variable *estadoDocumento* permite identificar si los documentos fueron despachados con éxito, asignados a un vehículo sin haberse despachado, o si nunca se asignó a algún vehículo para su despacho; de hecho, este último tipo se corrobora al observar que los documentos identificados con su estado efectivamente nunca fueron asignados a algún vehículo.

Por lo tanto, se puede concluir que aparentemente varios de los documentos emitidos por día no son despachados, lo que significa que, posiblemente esta variable pueda ser usada para un análisis de rendimiento en cuanto a comparaciones entre la planificación logística actual con las que LUMA podrá generar. La distribución de los documentos por su estado se puede visualizar a continuación:

Ilustración 7
Distribución de documentos por su estado



Sin embargo, tras conversaciones con los encargados del proyecto y de la empresa a la cual pertenecen los datos sobre estos hallazgos, se determinó que esta información es engañosa ya que en realidad todos los documentos emitidos por día son asignados para su despacho, es

decir, si bien en los registros se detectó que el 30.1% de todos los pedidos entre el 11 de marzo y 11 de abril fueron reportados como documentos sin asignar a ningún vehículo, esto solo evidencia que existen irregularidades al momento en el que el operador de logística asigna documentos a vehículos para su despacho.

Asimismo, adicional a las conversaciones mantenidas, durante una exploración con mayor detalle sobre los hallazgos presentados en la anterior ilustración, se identificó que los documentos sin asignar y asignados sin entregar, no vuelven a ser asignados en ninguna otra fecha, por lo que, provoca que surja el cuestionamiento sobre si los documentos asignados sin entregar corresponden realmente a documentos no entregados, pues, es posible que para este tipo de documentos se haya generado un nuevo código único para su identificación y despacho en una fecha diferente, por lo que, de ser así, resultaría imposible conocer cuáles son estas situaciones debido a que en muchas ocasiones los clientes realizan los mismos pedidos y todos tienen códigos de identificación diferentes.

De esta forma, teniendo presente estos hallazgos, implica que únicamente se pueda interpretar que los datos de los estados de asignado sin entregar y de sin asignar a algún vehículo, correspondan a datos que a lo mejor hayan sido exitosamente entregados pero que no se haya realizado su respectivo reporte; por lo tanto, se determinó que la variable *estadoDocumento* no es apta para usarla en análisis de rendimiento para comparar la planificación de logística actual con la que proveerá LUMA, no obstante, será útil una vez que el LUMA haya sido creada puesto que permitirá identificar qué documentos deben ser despachados y, también, reportar al sistema Practalis los documentos que ya han sido asignado para su despacho.

En este sentido, el resultado de la exploración de datos recién expuesto representa una muestra más de las deficiencias que puede ocasionar la planificación de logística manual,

dejando como resultado datos poco fiables para evaluar el rendimiento de una logística de última milla.

Por otro lado, continuando con la exploración del conjunto de datos, se construyó un nuevo dataframe agrupado por fecha con respecto al peso, volumen y cantidad de documentos emitidos, con el objetivo de explorar su comportamiento y variación a través del tiempo; de manera que, permitió que se descubra que en promedio se emiten 66 documentos por ser despachados por día, donde, en el 75% de estos días se emitieron menos de 74 documentos, habiéndose emitido 101 documentos como máximo, 5 como mínimo, y teniendo una desviación estándar de 20 documentos por día. Esta información puede ser observada con mayor detalle en las siguientes ilustraciones:

Ilustración 8

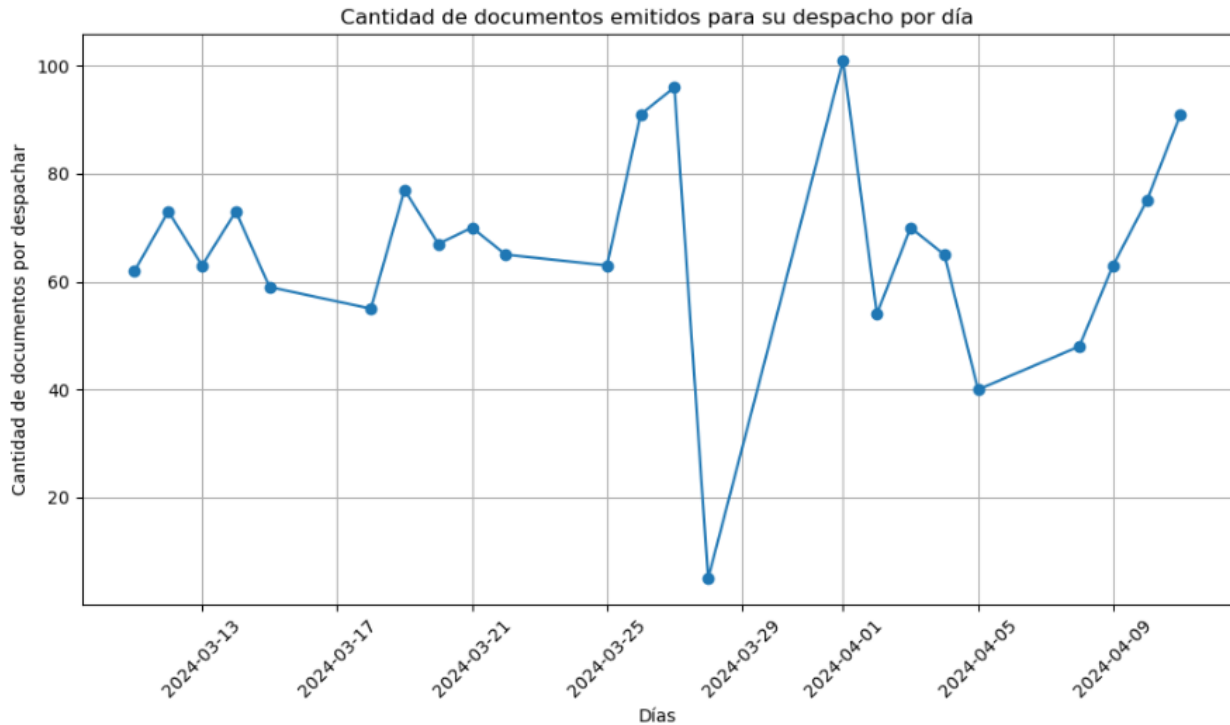
Estadísticas descriptivas de la cantidad de documentos que se emiten por día

```
# Se visualizan las estadísticas descriptivas de la variable cantidad
datos_grafico[["cantidad"]].describe()
```

| | cantidad |
|-------|------------|
| count | 23.000000 |
| mean | 66.347826 |
| std | 19.983195 |
| min | 5.000000 |
| 25% | 60.500000 |
| 50% | 65.000000 |
| 75% | 74.000000 |
| max | 101.000000 |

Ilustración 9

Cantidad de documentos emitidos para su despacho por día



En la misma línea, se puede observar que en las ilustraciones anteriores existe un día en el que la cantidad de documentos emitidos para su despacho es sumamente baja en comparación a los demás, por lo que, podría considerarse como un valor atípico que debe ser evaluado en el capítulo de evaluación de la calidad de los datos.

Por otra parte, a pesar de que pueda resultar intuitivo suponer que los días con mayor cantidad de documentos implica que se deberá despachar una mayor cantidad de peso o volumen, no es del todo cierto, ya que, como se puede observar en las dos ilustraciones a continuación, la cantidad de peso y volumen por ser despachada por día no sigue esta lógica estrechamente.

Ilustración 10

Cantidad de documentos emitidos para su despacho y el peso total por despachar por día

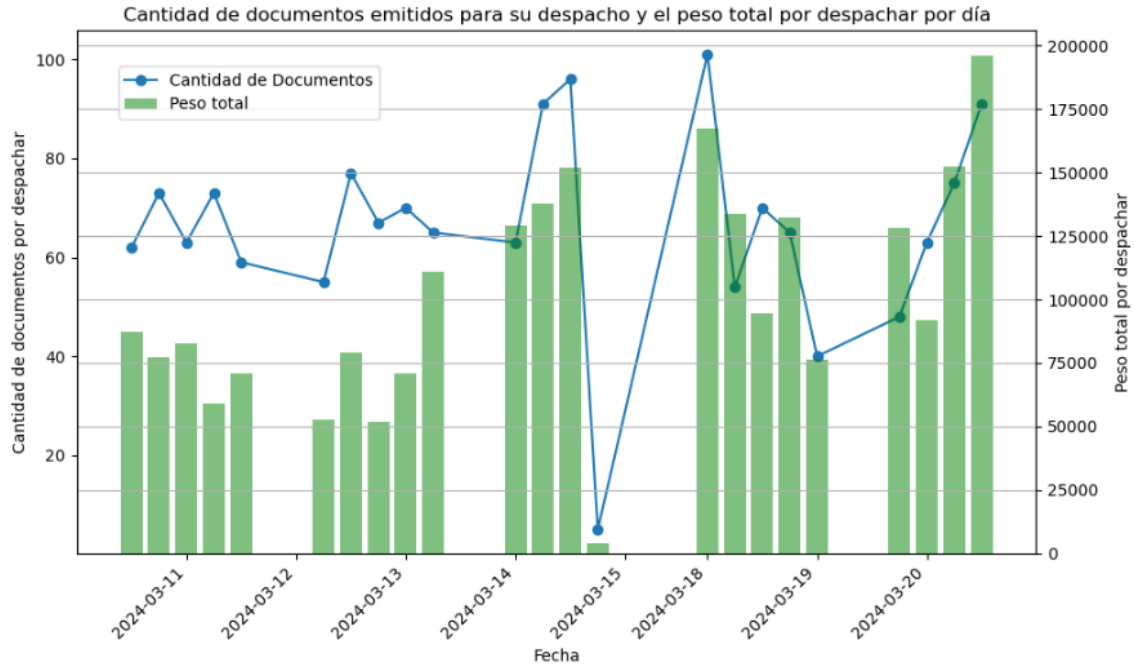
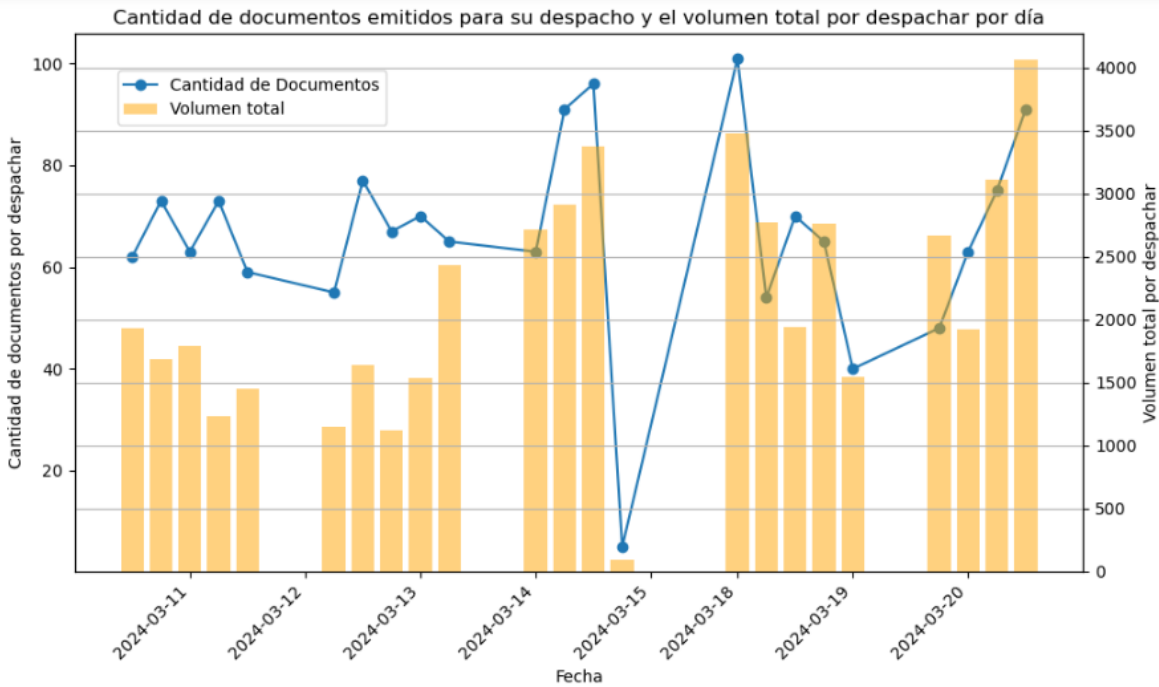


Ilustración 11

Cantidad de documentos emitidos para su despacho y el volumen total por despachar por día



Esto se debe a que un solo documento puede corresponder a varias mercancías, es decir, un documento puede equivaler al peso de dos mercancías mientras que otro al de diez y así sucesivamente.

Finalmente, concluyendo la exploración de datos de Documentos, se identificó que varias variables poseen valores constantes, por lo que, en otras palabras, podría significar que no presentarán utilidad alguna para el desarrollo de LUMA, sin embargo, es necesario evaluarlas en el capítulo de evaluación de la calidad de los datos.

Ilustración 12

Listado de variables con valores constantes

```
# Se enlistan las variables con valores constantes
variables_con constantes = [
    "estadoProceso",
    "estadoProcesoDetalle",
    "seriePuntoVenta",
    "serieCaja",
    "estadoProcesoActividad",
    "nombreFacilidad",
    "nombreCaja"
]
```

Ilustración 13

Estadísticas descriptivas de variables con valores constantes

```
# Se visualizan las estadísticas descriptivas de las variables numéricas  
# con valores constantes  
documentos[variables_con_constantes].describe()
```

| | estadoProceso | estadoProcesoDetalle | seriePuntoVenta | serieCaja |
|-------|---------------|----------------------|-----------------|-----------|
| count | 1602.0 | 1602.0 | 1602.0 | 1602.0 |
| mean | 2.0 | 0.0 | 1.0 | 5.0 |
| std | 0.0 | 0.0 | 0.0 | 0.0 |
| min | 2.0 | 0.0 | 1.0 | 5.0 |
| 25% | 2.0 | 0.0 | 1.0 | 5.0 |
| 50% | 2.0 | 0.0 | 1.0 | 5.0 |
| 75% | 2.0 | 0.0 | 1.0 | 5.0 |
| max | 2.0 | 0.0 | 1.0 | 5.0 |

```
# Se visualizan las estadísticas descriptivas de las variables categóricas  
# con valores constantes  
documentos[variables_con_constantes].describe(include=object)
```

| | estadoProcesoActividad | nombreFacilidad | nombreCaja |
|--------|------------------------|-----------------|------------|
| count | 1602 | 1602 | 1602 |
| unique | 1 | 1 | 1 |
| top | A | OFICINA MATRIZ | LOGISTICA |
| freq | 1602 | 1602 | 1602 |

5.3.2. Hallazgos de la exploración de los datos de Vehículos

Con respecto al conjunto de datos de Vehículos, se identificó que similarmente existen variables que poseen valores con información constante que deberán ser evaluadas durante la verificación de los datos, estas son:

Ilustración 14

Estadísticas descriptivas de las variables con valores constantes del conjunto de datos de Vehículos

```
# Se enlistan las variables con valores constantes
variables_con constantes = [
    "capacidadUnidad",
    "volumenUnidad",
    "estadoVehiculo"
]
# Se visualización las estadísticas descriptivas de las variables
# con valores constantes
vehiculos[variables_con constantes].describe()
```

| | capacidadUnidad | volumenUnidad | estadoVehiculo |
|--------|-----------------|---------------|----------------|
| count | 9 | 9 | 9 |
| unique | 1 | 1 | 1 |
| top | KG | QQ | A |
| freq | 9 | 9 | 9 |

Por otro lado, también se identificó un error en los datos de la variable *ultimoAnioMatricula*, en el cual sus registros contienen el año 2021 por excepción de uno con el valor de “2200”, por lo cual, es igualmente necesario analizarlo en el capítulo de verificación de la calidad de los datos.

Ilustración 15

Visualización de las categorías de la variable *ultimoAnioMatricula*

```
# Visualización de las categorías de la variable "ultimoAnioMatricula"
vehiculos[["ultimoAnioMatricula"]]
```

| | ultimoAnioMatricula |
|---|---------------------|
| 0 | 2021 |
| 1 | 2021 |
| 2 | 2021 |
| 3 | 2021 |
| 4 | 2021 |
| 5 | 2021 |
| 6 | 2021 |
| 7 | 2021 |
| 8 | 2200 |

5.3.3. Hallazgos de la exploración de los datos integrados

Para empezar, debido a la existencia de variables que tienen el objetivo de, tras su concatenación, identificar diferentes actores del proceso de logística de última milla como, por ejemplo, los documentos, despachadores, vehículos y clientes; es imprescindible que el primer paso llevado a cabo para la exploración de los datos integrados de Documentos y Vehículos sea la creación de dichas variables identificadoras, debido a que, gracias a esto se permitirá dicha integración. Es por esto por lo que se creó una función que permitirá crearlas a partir del listado de las variables que las componen, la función es la siguiente:

Ilustración 16

Función para la creación de variables identificadoras

```
def crear_variable_identificadora(df, nombre_nuevo, variables_por_concatenar):  
    ...  
    - La función agrega una nueva columna al dataframe considerado en df.  
    - El nombre de la nueva columna será el considerado en nombre_nuevo.  
    - Los datos que corresponderán a la nueva columna serán los de la concatenación  
      de las variables enlistadas en variables_por_concatenar.  
    ...  
    df[nombre_nuevo] = df[variables_por_concatenar].astype(str).agg("".join, axis=1)
```

Posteriormente, se identificaron y enlistaron las variables requeridas para la creación de las variables identificadoras de los distintos actores que forman parte del proceso de logística de última milla, estos son: los documentos, despachadores, vehículos, clientes y las direcciones de estos últimos.

Ilustración 17

Identificación de variables requeridas para la creación de variables identificadoras en el conjunto de datos de Documentos

```
...
Se identifican las variables requeridas para la creación de variables identificadoras.
...
# Variables para la identificación de documentos
variables_id_documento = [
    "codigoEmpresa",
    "codigoFacilidad",
    "codigoCaja",
    "codigoTipoDocumento",
    "codigoSerieDocumento",
    "codigoDocumento"
]
# Variables para la identificación de despachadores
variables_id_despachador = [
    "codigoEmpresa",
    "codigoDespachador",
    "sucursalDespachador"
]
# Variables para la identificación de vehículos
variables_id_vehiculo = [
    "codigoEmpresa",
    "codigoFacilidad",
    "codigoVehiculo"
]
# Variables para la identificación de clientes
variables_id_cliente = [
    "codigoEmpresa",
    "codigoTercero",
    "sucursalTercero"
]
# Variables para la identificación de las direcciones de los clientes
variables_id_direccion_cliente = [
    "codigoEmpresa",
    "codigoTercero",
    "sucursalTercero",
    "codigoTerceroDireccion"
]
]
```

Finalmente, se procedió con la creación de las variables identificadoras mediante el uso de la función creada anteriormente, para que, después, se proceda con la eliminación de las variables que fueron concatenadas para sus respectivas creaciones.

Ilustración 18

Creación de variables identificadores y eliminación de variables usadas para ello

```
...
Se crean las variables identificadores y posteriormnete, se eliminan las variables que
fueron usadas para su creación.
...
# Creación de variables identificadoras en el conjunto de datos de Documentos
crear_variable_identificadora(documentos, "id_documento", variables_id_documento)
crear_variable_identificadora(documentos, "id_despachador", variables_id_despachador)
crear_variable_identificadora(documentos, "id_vehiculo", variables_id_vehiculo)
crear_variable_identificadora(documentos, "id_cliente", variables_id_cliente)
crear_variable_identificadora(documentos, "id_direccion_cliente", variables_id_direccion_cliente)

# Creación de variables identificadoras en el conjunto de datos de Vehículos
crear_variable_identificadora(vehiculos, "id_vehiculo", variables_id_vehiculo)

# Eliminación de variables en el conjunto de datos de Documentos
variables_por_eliminar = set(
    variables_id_documento +
    variables_id_despachador +
    variables_id_vehiculo +
    variables_id_cliente +
    variables_id_direccion_cliente
)
documentos.drop(columns=variables_por_eliminar, inplace=True)

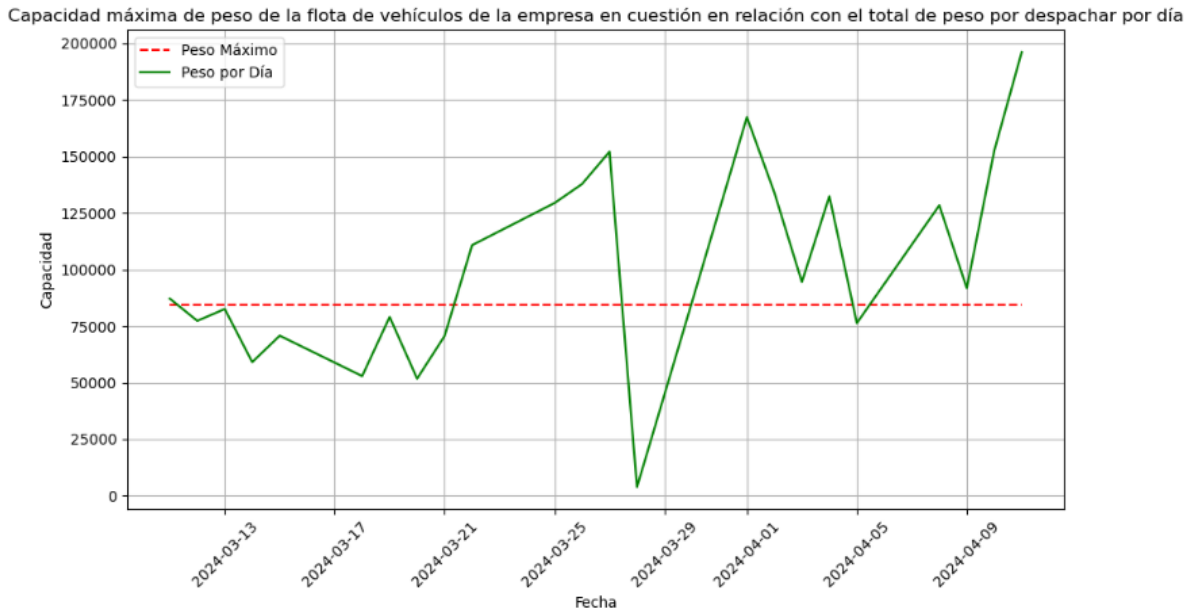
# Eliminación de variables en el conjunto de datos de Vehículos
vehiculos.drop(columns=variables_id_vehiculo, inplace=True)
```

De esta forma, se ha logrado adquirir variables que permiten la identificación de los actores que forman parte del proceso de logística de última milla en el conjunto de datos de Documentos y Vehículos, de manera que, es factible realizar su integración y exploración, sin embargo, es importante notar que este procedimiento no influirá sobre la preparación de los datos, pues, en ella se partirá desde la creación de las variables identificadoras mas no de la integración de datos que se realizará en este momento, es decir, este nuevo dataframe será únicamente utilizado para fines exploratorios.

De tal modo que, habiendo integrado los conjuntos de datos de Documentos y Vehículos, se comenzará su exploración a partir de la siguiente ilustración:

Ilustración 19

Capacidad máxima de peso de la flota de vehículos de la empresa en cuestión en relación con el total de peso por despachar por día

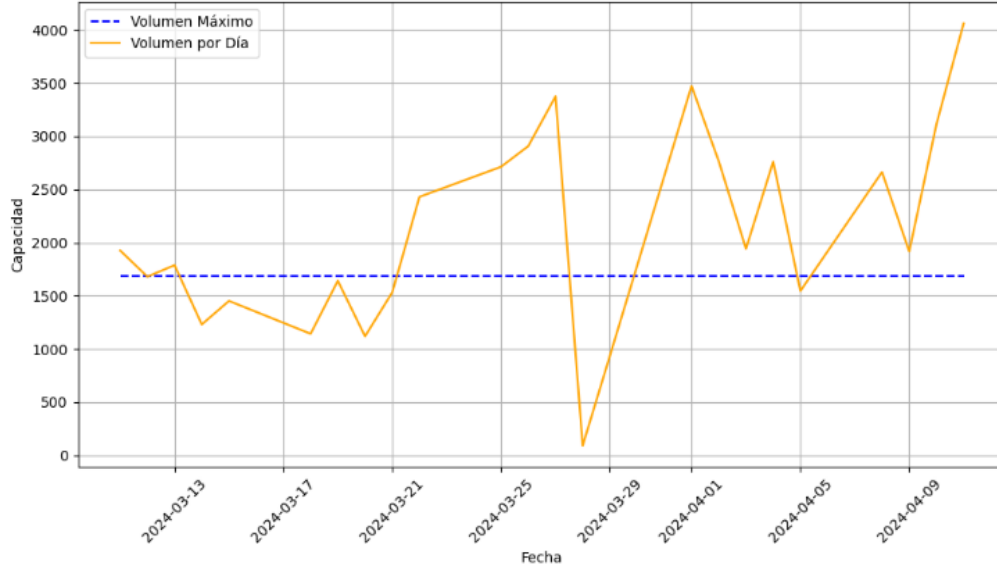


En la ilustración se pueden observar dos líneas, la roja entrecortada representa la capacidad máxima que pueden abarcar, en conjunto, todos los vehículos en cuanto a peso, mientras que, la línea verde continua representa el total del peso por ser despachado por día; lo que hace evidente la razón por la cual los vehículos deben realizar más de una ruta para poder concretar el despacho de todos los documentos que se emiten por día. De la misma forma ocurre con respecto a los volúmenes de los vehículos y documentos, como se puede observar a continuación:

Ilustración 20

Capacidad máxima de volumen de la flota de vehículos de la empresa en cuestión en relación con el total de volumen por despachar por día

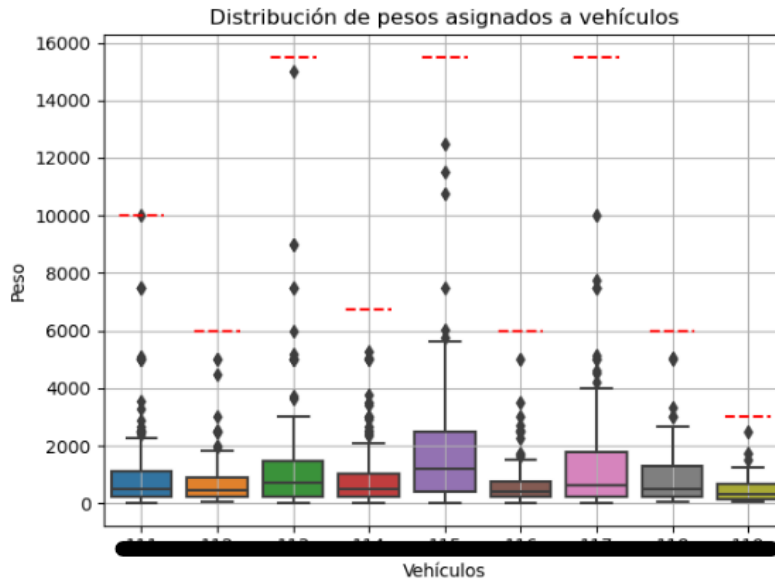
Capacidad máxima de volumen de la flota de vehículos de la empresa en cuestión en relación con el total de volumen por despachar por día



No obstante, adicional a que los vehículos deben realizar más de una ruta para completar el despacho de todos los documentos, durante la planificación de logística de última milla manual, no se ha utilizado de manera eficiente su capacidad, pues, como se puede observar en la siguiente ilustración, donde, la línea roja entrecortada simboliza la capacidad máxima de cada vehículo en cuanto a peso y los diagramas de caja y bigote la distribución de los pesos que se les ha sido asignados a cada uno; refleja claramente que existe una subutilización de recursos en cuanto a las capacidades de los vehículos.

Ilustración 21

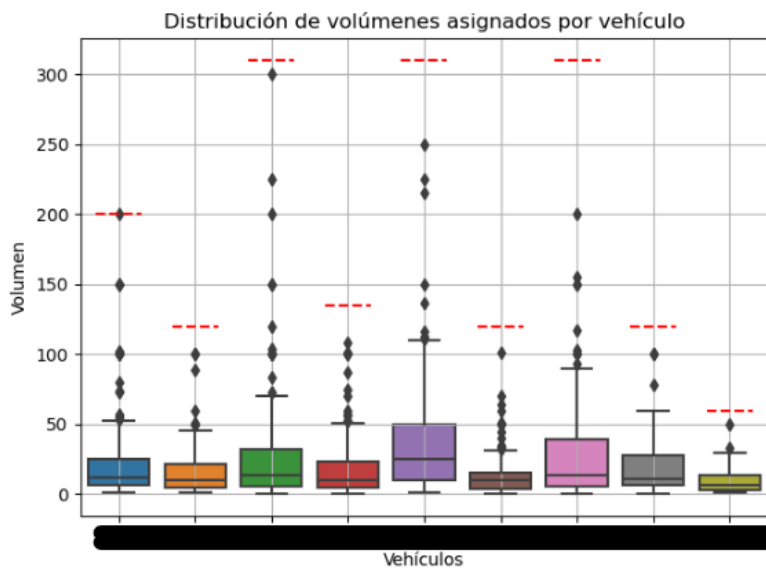
Distribución de pesos asignados a vehículos



Además, esto no se limita únicamente con respecto a los pesos, ya que, de igual forma sucede con respecto a la distribución de volúmenes en los vehículos; eventualidad que puede ser observada a continuación:

Ilustración 22

Distribución de volúmenes asignados por vehículo

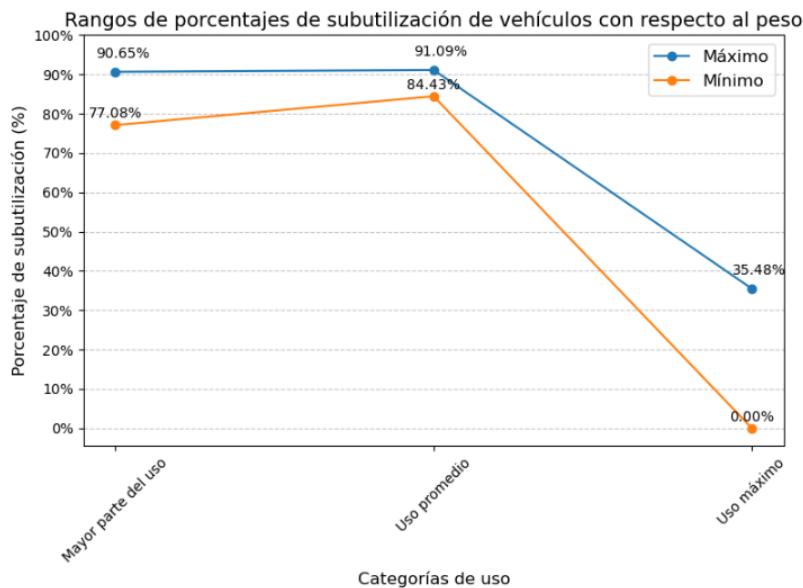


En este sentido, se evidencia que los vehículos en la mayoría de las ocasiones están siendo subutilizados en términos de la capacidad máxima del peso y volumen que pueden cargar, de hecho, considerando el 75% de los datos, es decir, en la mayoría de las ocasiones que se han asignado documentos a vehículos, ha existido un porcentaje de subutilización de su capacidad con relación al peso de entre el 77.08% hasta el 90.65% dependiendo del vehículo, mientras que, en promedio, este porcentaje asciende a un valor mínimo de 84.43% hasta un máximo de 91.09%, no obstante, en términos del uso máximo que se le han dado a estos vehículos, el resultado es mucho más optimista pues los porcentajes de subutilización se reducen a valores desde un 0% hasta un 35.48%, lo que, de igual manera abre espacio para su optimización.

Estos rangos de porcentajes pueden ser observados a detalle en el siguiente gráfico:

Ilustración 23

Rangos de porcentajes de subutilización de vehículos con respecto al peso



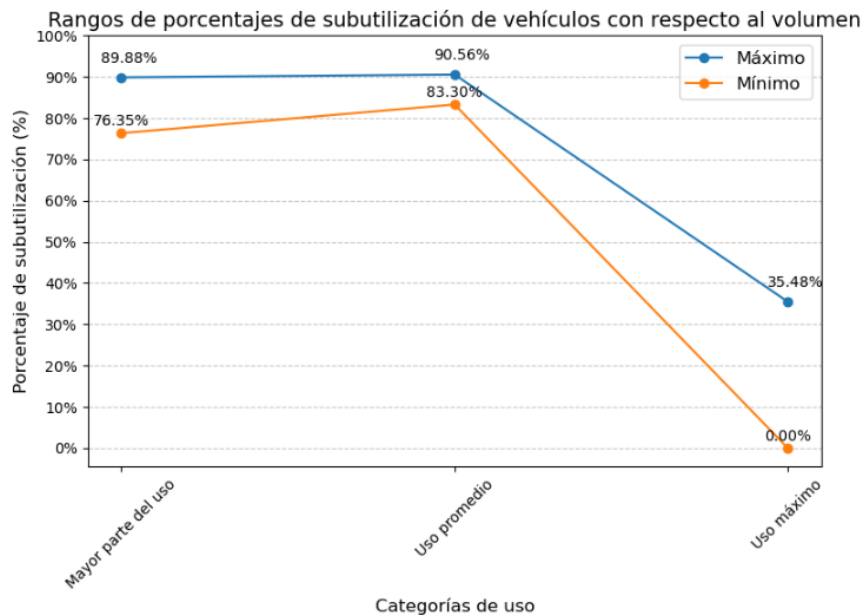
Asimismo, de manera similar que con el peso, considerando el 75% de los datos, es decir, en la mayoría de las ocasiones que se han asignado documentos a vehículos, ha existido un porcentaje de subutilización de su capacidad con relación al volumen de entre el 76.35% hasta el

80.88% dependiendo del vehículo, mientras que, en promedio, este porcentaje asciende a un valor mínimo de 83.30% hasta un máximo de 90.56%, no obstante, en términos del uso máximo que se le han dado a estos vehículos, el resultado es idéntico al de los pesos, porque los porcentajes de subutilización se reducen a valores desde un 0% hasta un 35.48%, lo que, similarmente abre espacio para su optimización.

Estos rangos de porcentajes con respecto al volumen pueden ser observados a detalle en el siguiente gráfico:

Ilustración 24

Rangos de porcentajes de subutilización de vehículos con respecto al volumen



Finalmente, también se evidencio la presencia de la anomalía definida en el alcance del trabajo, la cual sucede cuando un solo documento posee un mayor peso o volumen que el vehículo con mayor capacidad en ambos contextos. En las siguientes dos ilustraciones se puede observar cuándo ocurren estas anomalías.

Ilustración 25

Capacidad máxima del vehículo con mayor capacidad con respecto al peso en relación con el documento con mayor peso por despachar por día

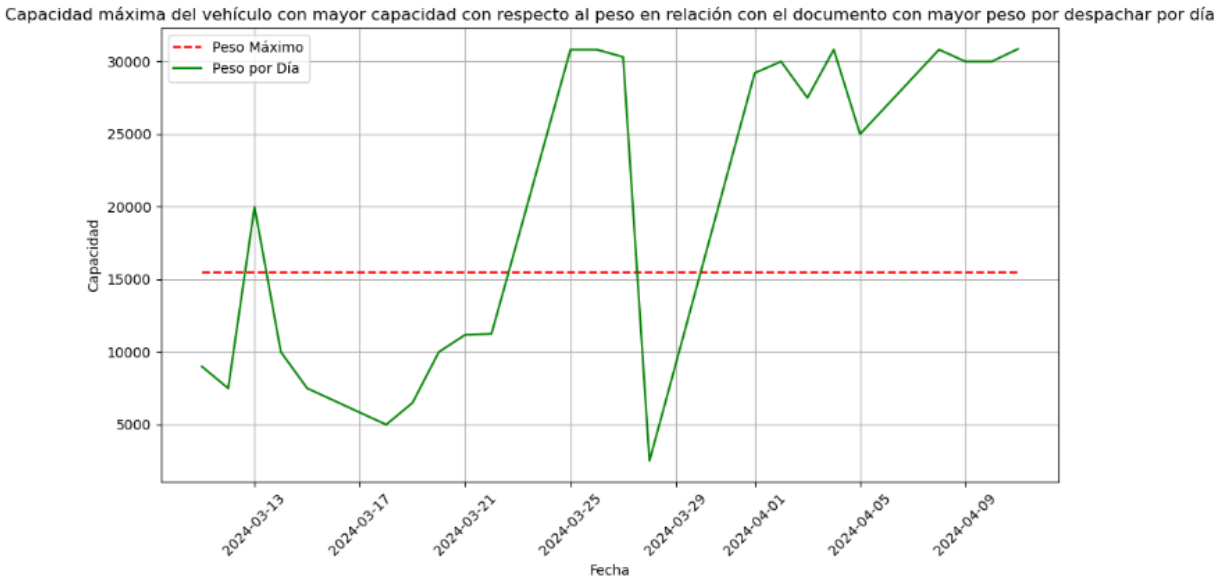
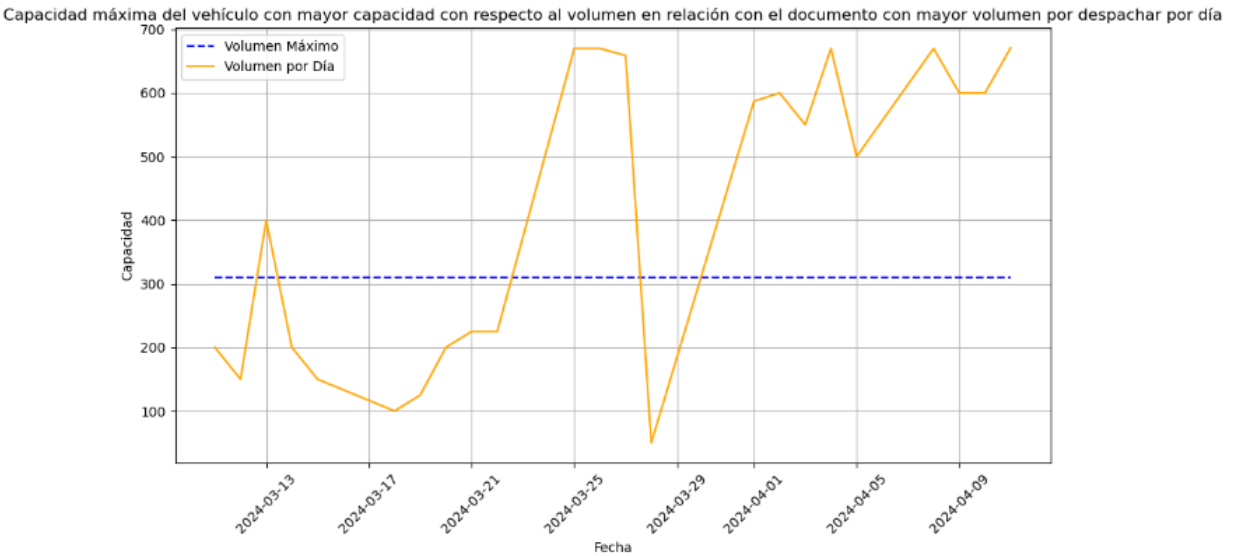


Ilustración 26

Capacidad máxima del vehículo con mayor capacidad con respecto al volumen en relación con el documento con mayor volumen por despachar por día



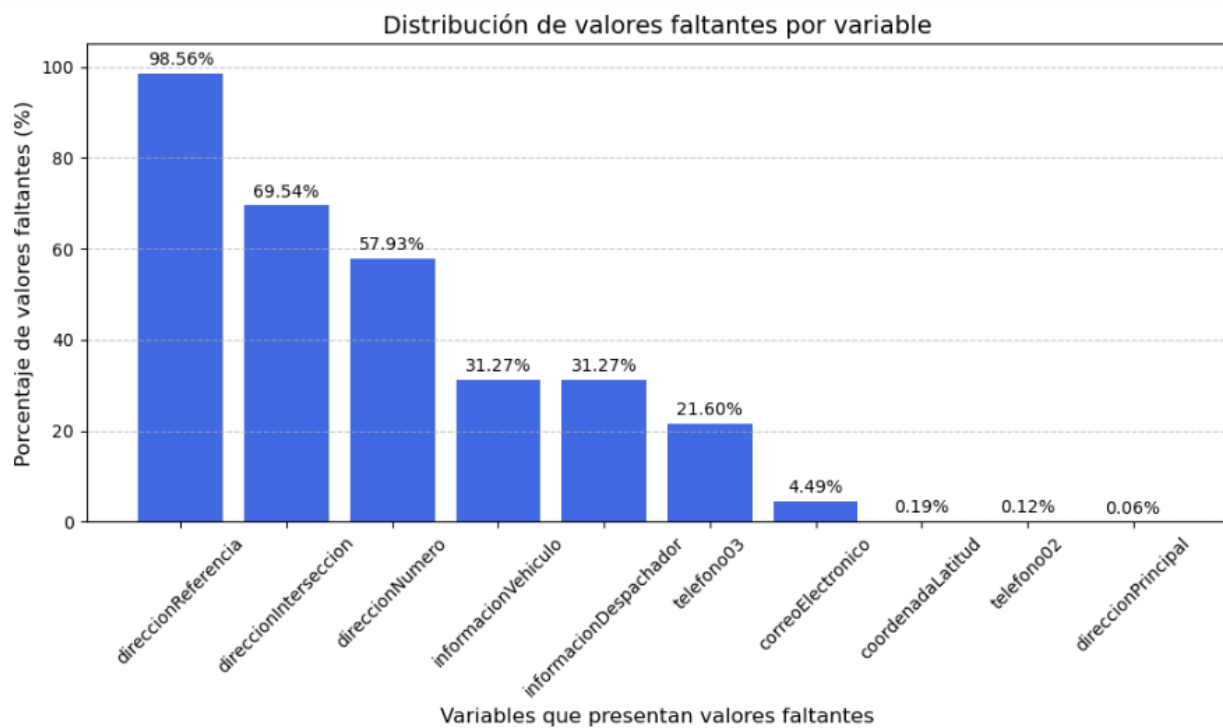
5.4. Verificación de la calidad de los datos

Para empezar, con respecto al conjunto de datos de Documentos, tal como se evidencia en la ilustración 27, existe la presencia de valores faltantes distribuidos a lo largo de diez de sus variables.

Sin embargo, a pesar de que en algunos casos sus porcentajes de valores nulos ascienden hasta un aproximado del 98%, estas ausencias no representan un problema debido a que la mayor parte de dichas variables pertenecen a información que no presenta utilidad para el desarrollo de LUMA, más bien, en su mayoría presenta información personal identificable.

Ilustración 27

Distribución de valores faltantes por variable



No obstante, existen casos que generan preocupación como la presencia de valores faltantes en la variable *coordenadaLatitud*, debido a que, al ser una variable imprescindible para la localización de los puntos de entrega de los documentos que se emiten para su despacho,

podría ocasionar problemas pues no hubo forma de conocer dicha información ni posibilidad de insertarla ya que sería antinatural.

De todas maneras, debido a que su porcentaje de valores faltantes es únicamente del 0.19% y que, en realidad, durante la planificación de la logística de última milla que se lleva actualmente a cabo en Practalis, los documentos que no poseen su información geográfica completa son descartados de la planificación de su despacho, entonces, se puede optar por la eliminación de los registros con dichos valores faltantes del conjunto de datos de Documentos.

Por otra parte, se descubrió que algunas de las coordenadas de los documentos poseen latitudes y longitudes iguales a 0, y, si bien es algo factible considerando que Ecuador tiene la posibilidad de poseer ambas coordenadas, al observar las direcciones principales de dichos documentos y averiguar sus coordenadas reales, se pudo destacar que los valores registrados en los datos no corresponden a los reales provocando que se pueda determinar que es un error, por lo tanto, se puede optar por la eliminación de dichos registros.

Ahora, en cuanto a las variables con información constante en sus datos que fueron identificadas durante la exploración de datos, se sabe que en el conjunto de datos de documentos existen 7 ocurrencias, mientras que, en el conjunto de datos de Vehículos, existe un total de 3. Esta información puede ser vuelta a analizar en las ilustraciones a continuación:

Ilustración 28

Estadísticas descriptivas de variables con valores constantes del conjunto de datos de Documentos

```
# Se visualizan las estadísticas descriptivas de las variables numéricas  
# con valores constantes  
documentos[variables_con_constantes].describe()
```

| | estadoProceso | estadoProcesoDetalle | seriePuntoVenta | serieCaja |
|-------|---------------|----------------------|-----------------|-----------|
| count | 1602.0 | 1602.0 | 1602.0 | 1602.0 |
| mean | 2.0 | 0.0 | 1.0 | 5.0 |
| std | 0.0 | 0.0 | 0.0 | 0.0 |
| min | 2.0 | 0.0 | 1.0 | 5.0 |
| 25% | 2.0 | 0.0 | 1.0 | 5.0 |
| 50% | 2.0 | 0.0 | 1.0 | 5.0 |
| 75% | 2.0 | 0.0 | 1.0 | 5.0 |
| max | 2.0 | 0.0 | 1.0 | 5.0 |

```
# Se visualizan las estadísticas descriptivas de las variables categóricas  
# con valores constantes  
documentos[variables_con_constantes].describe(include=object)
```

| | estadoProcesoActividad | nombreFacilidad | nombreCaja |
|--------|------------------------|-----------------|------------|
| count | 1602 | 1602 | 1602 |
| unique | 1 | 1 | 1 |
| top | A | OFICINA MATRIZ | LOGISTICA |
| freq | 1602 | 1602 | 1602 |

Ilustración 29

Estadísticas descriptivas de las variables con valores constantes del conjunto de datos de Vehículos

```
# Se enlistan las variables con valores constantes  
variables_con_constantes = [  
    "capacidadUnidad",  
    "volumenUnidad",  
    "estadoVehiculo"  
]  
# Se visualización las estadísticas descriptivas de las variables  
# con valores constantes  
vehiculos[variables_con_constantes].describe()
```

| | capacidadUnidad | volumenUnidad | estadoVehiculo |
|--------|-----------------|---------------|----------------|
| count | 9 | 9 | 9 |
| unique | 1 | 1 | 1 |
| top | KG | QQ | A |
| freq | 9 | 9 | 9 |

De esta forma, habiendo analizado la naturaleza de los datos y conversado con el encargado de los datos de Biologicsoft, se logró concluir los siguientes aspectos por cada variable:

Tabla 18
Análisis de variables con valores constantes

| Variable | Conjunto de datos | Análisis |
|------------------------|-------------------|--|
| estadoProceso | Documentos | La variable identifica cuando un documentos está confirmado para su despacho, por lo que, no es de utilidad ya que todos los documentos que se enviarán a LUMA estarán previamente ya confirmados. |
| estadoProcesoDetalle | Documentos | Esta variable indica cuando un documento está autorizado para su despacho, por ende, al considerar que todos los documentos siempre estarán autorizados antes de ser enviados a LUMA, entonces, no presenta utilidad. |
| seriePuntoVenta | Documentos | Es la serie del establecimiento, sin embargo, en todas las ocasiones siempre será un valor constante, además, no presenta utilidad para a los objetivos propuestos. |
| serieCaja | Documentos | Esta variable representa la serie de la caja donde se realizó la compra, sin embargo, aplica lo mismo que para la variable <i>seriePuntoVenta</i> . |
| estadoProcesoActividad | Documentos | La variable identifica el subproceso de facturación en el que se encuentra el documento, sin embargo, no se ha llevado a cabo un buen seguimiento de su información, por lo tanto, no presenta utilidad. |
| nombreFacilidad | Documentos | La variable contiene el nombre del establecimiento de la facilidad de donde saldrán los vehículos de despacho, por lo tanto, no presenta utilidad porque independientemente de la empresa, LUMA considerará un único punto de origen de los vehículos. |
| nombreCaja | Documentos | Sus datos corresponden al nombre de la caja donde se realizó la venta y aplica lo mismo que para la variable <i>serieCaja</i> . |
| capacidadUnidad | Vehículos | Es la unidad de medida de la capacidad con respecto al peso del vehículo, por lo que, la poseer las mismas medidas que el conjunto de datos de Documentos, no presenta utilidad. |

| | | |
|----------------|-----------|---|
| volumenUnidad | Vehículos | Corresponde a la capacidad total del vehículo con respecto al volumen y aplica lo mismo que para la variable <i>capacidadUnidad</i> . |
| estadoVehiculo | Vehículos | Corresponde al estado en el que se encuentra el vehículo, es decir, si se encuentra disponible para realizar despachos o si se encuentra realizándolos, por lo que, resulta útil para que LUMA pueda seleccionar que vehículos serán considerados para la planificación automática de la logística de última milla. |

Por otro lado, igualmente basándose en los hallazgos de la exploración de datos, se identificó un posible valor atípico que consistía en que en una fecha la cantidad de documentos emitidos para su despacho fue de 5 cuando normalmente es de 66, no obstante, a pesar de que los registros sugieren que los documentos no fueron asignados a algún vehículo y por lo tanto no fueron despachados, debido a que los resultados de la exploración de datos con respecto a este tema y a las conversaciones con los encargados del proyecto y de la empresa a la cual pertenecen los datos, se debe considerar a estos registros como no atípicos, en otras palabras, es posible que en algún día la cantidad de documentos por ser despachados varíe considerablemente, por lo que, LUMA deberá ser capaz de planificar la logística de última milla para cualquier cantidad de documentos que se emitan en el día.

Finalmente, tras haber realizado una exploración y verificación de los datos, se puede concluir que los conjuntos de datos de Documentos y Vehículos son adecuados para la creación y desarrollo de LUMA.

CAPÍTULO VI: PREPARACIÓN DE DATOS

6. Preparación de datos

Una vez finalizada la exploración de los datos y la verificación de su calidad, ha llegado el momento en el que, con base a sus descubrimientos, se prepararen los conjuntos de datos de Documentos y Vehículos para poder desarrollar LUMA, es decir, este capítulo tiene el objetivo de seleccionar las variables relevantes para el desarrollo y creación de LUMA, limpiar sus datos, formatearlos y estructurarlos de tal manera que sea fácil e intuitivo interactuar con ellos.

Por lo que, para comenzar se partirá con la identificación de variables que serán eliminadas y mantenidas en cada conjunto de datos:

6.1. Identificación de variables para la selección de datos

Para comenzar, se debe mencionar que la evaluación llevada a cabo para la selección de esta fase fue realizada en conjunto con la exploración de datos en donde se determinó que algunas variables, si bien, pueden presentar información útil para distintas etapas del proceso de logística de última milla, para el desarrollo concreto de LUMA no presenta utilidad.

Es por esto por lo que, a raíz de la evaluación realizada para identificar dicha información, a continuación, se podrán observar tablas en las que se especifican cuáles van a ser las variables que serán descartadas junto con sus respectivos motivos.

Primero, partiendo de la evaluación llevada a cabo sobre las variables del conjunto de datos de Documentos, se concluyó prudente descartar las siguientes:

Tabla 19
Variables por descartar del conjunto de datos de Documentos

| Variable | Motivos |
|-----------------|----------------|
|-----------------|----------------|

| | |
|------------------------|---|
| fechaElaboracion | Esta variable no presenta utilidad debido a que no es importante el cuándo se emitió el documento por ser despachado, sino, el cuándo se solicitó su entrega. |
| fechaVencimiento | Esta variable representa únicamente la fecha máxima de pago del cliente por la mercancía solicitada, por lo que, no presenta utilidad. |
| totalDocumento | Esta variable corresponde al valor neto del documento, por ello, debido a que la premisa es la entrega de todos los documentos en el día y tampoco se tiene información sobre ventanas horarias, la variable no tiene utilidad. |
| informacionVehiculo | Esta variable contiene información ya disponible en el conjunto de datos de Vehículos, y, además, presenta información no útil con respecto a detalles del vehículo como su modelo y marca. |
| informacionDespachador | Esta variable presenta el nombre del despachador asignado a los vehículos, por lo que, no resulta útil para el desarrollo de LUMA. |
| estadoProceso | Esta variable a pesar de que represente el estado de un documento en cuanto a su confirmación para ser despachado no es útil debido a que, por defecto, todos los documentos que reciba LUMA estarán confirmados. |
| estadoProcesoDetalle | Esta variable a pesar de que represente el estado de un documento en cuanto a su autorización para ser despachado no es útil debido a que, por defecto, todos los documentos que reciba LUMA estarán autorizados. |
| estadoProcesoActividad | Esta variable no es útil debido a que esta información es indeterminada en cuando a todos los estados de facturación en los que el documento realmente se encuentre. |
| codigoUnificado | Esta variable al ser parte de un código utilizado para la sincronización del sistema Practalis Integrado con otro tercero, no presenta utilidad para el desarrollo concreto de LUMA. |
| codigoUnificado02 | Esta variable al ser parte de un código utilizado para la sincronización del sistema Practalis Integrado con otro tercero, no presenta utilidad para el desarrollo de la tesis. |
| nombreEmpresaTercero | Esta variable puede ser descartada ya que únicamente desvela información personal identificable de quienes solicitan el despacho de sus mercancías. |
| nombreComercialTercero | Esta variable puede ser descartada ya que únicamente desvela información personal identificable de quienes solicitan el despacho de sus mercancías. |
| tipoIdentificacion | Esta variable corresponde al tipo de identificación utilizado por quienes solicitan el despacho de sus mercancías, por lo que, no es relevante. |

| | |
|-------------------------|---|
| documentoIdentificacion | Esta variable puede ser descartada ya que únicamente desvela información personal identificable de quienes solicitan el despacho de sus mercancías. |
| direccionPrincipal | Esta variable no presenta utilidad puesto que la localización de los puntos de entrega de los documentos se realizará mediante sus coordenadas de longitud y latitud. |
| direccionInterseccion | Esta variable se ha considerado inútil debido a que el 69.54% de sus datos son valores faltantes, y, además, su información forma parte de los datos de la variable <i>direccionPrincipal</i> . |
| direccionNumero | Esta variable solo presenta utilidad una vez se haya arribado al punto de entrega del documento, de manera que, no es relevante para el desarrollo de LUMA. |
| direccionReferencia | Esta variable, además de tener un 98.56% de valores faltantes, no desvela información útil para el desarrollo del trabajo. |
| telefono01 | Esta variable puede ser descartada pues únicamente desvela información personal identificable de quienes solicitan el despacho de sus mercancías. |
| telefono02 | Esta variable puede ser descartada pues únicamente desvela información personal identificable de quienes solicitan el despacho de sus mercancías. |
| telefono03 | Esta variable puede ser descartada pues únicamente desvela información personal identificable de quienes solicitan el despacho de sus mercancías. |
| correoElectronico | Esta variable puede ser descartada pues únicamente desvela información personal identificable de quienes solicitan el despacho de sus mercancías. |
| nombreFacilidad | Esta variable no es relevante debido a que, con base al alcance de LUMA, siempre se considerará un único punto de origen para los vehículos de despacho. |
| seriePuntoVenta | Esta variable es la representación en código de <i>nombreFacilidad</i> , por lo que, no es relevante. |
| nombreCaja | Esta variable no presenta utilidad puesto que en todos los casos se corresponde a la misma caja. |
| serieCaja | Esta variable es la representación en código de <i>nombreCaja</i> , por lo que, tampoco es relevante. |
| id_documento | Esta variable fue creada durante la exploración de datos, sin embargo, no presenta utilidad para desarrollos posteriores de LUMA debido a que se identificó que todos sus registros son y serán únicos. |

| | |
|----------------------|--|
| id_cliente | Esta variable fue creada durante la exploración de datos, sin embargo, al únicamente identificar a quienes solicitan los documentos, resulta irrelevante su información puesto que las coordenadas geográficas ya permiten el trabajo de identificar los puntos de entrega y cada registro permite resaltar su individualidad. |
| id_despachador | Esta variable igualmente fue creada durante la exploración de datos, sin embargo, al únicamente identificar a los conductores de los vehículos de despacho, no presenta utilidad. |
| id_direccion_cliente | Esta variable similarmente fue creada durante la exploración de datos, no obstante, al identificar la dirección del cliente cuando ya se poseen sus coordenadas, resulta irrelevante. |

Por otro lado, con respecto al conjunto de datos de Vehículos, se resolvió descartar las siguientes variables:

Tabla 20
Variables por descartar del conjunto de datos de Vehículos

| Variable | Motivos |
|---------------------|--|
| marcaVehiculo | Esta variable presenta la marca del vehículo, de manera que, para desarrollo de LUMA no presenta utilidad. |
| modeloVehiculo | Esta variable presenta el modelo del vehículo, por lo tanto, para el desarrollo del trabajo no presenta utilidad. |
| anioVehiculo | Esta variable corresponde al año de producción del vehículo, por lo cual, dado que no existen restricciones ni limitaciones con base al uso de los vehículos de despacho con relación a su longevidad, la variable resulta irrelevante para el desarrollo de LUMA. |
| numeroMatricula | Esta variable no presenta utilidad debido a que representa información para la identificación de la matrícula del vehículo, por lo que, no es significativa para el desarrollo de la tesis. |
| numeroPlacas | Esta variable identifica las placas de los vehículos de despacho, por esta razón, teniendo presente que no se han especificado limitaciones con respecto a su uso para el alcance del trabajo, la variable resulta irrelevante. |
| ultimoAnioMatricula | Esta variable no presenta utilidad debido a que representa información sobre la matrícula, que, para efecto del desarrollo de LUMA, no presenta utilidad. |
| numeroChasis | La variable corresponde al número del chasis, por lo que, claramente no presenta utilidad para el desarrollo de LUMA. |

| | |
|-----------------------|---|
| capacidadUnidad | Esta variable únicamente presenta la unidad de medida de la capacidad del vehículo en cuanto al peso, por lo que, debido a que todos los datos de pesos siguen la misma unidad, no existe necesidad de usarla. |
| volumenUnidad | Esta variable únicamente presenta la unidad de medida de la capacidad del vehículo en cuanto al volumen, por lo que, debido a que todos los datos de volúmenes siguen la misma unidad, no existe necesidad de usarla. |
| tieneRastreoVehicular | Esta variable identifica si el vehículo de despacho cuenta con rastreo vehicular, sin embargo, dado el alcance del trabajo, la variable no será considerada. |
| tieneSeguro | Esta variable identifica si el vehículo de despacho cuenta con seguro, sin embargo, dado el alcance del trabajo, la variable no será considerada. |
| codigoUnificado | Esta variable al ser parte de un código utilizado para la sincronización del sistema Practalis Integrado con otro tercero, no presenta utilidad para el desarrollo de la tesis. |

6.2. Selección de variables

Una vez identificadas las variables que no presentan utilidad para el desarrollo del trabajo, es momento de descartarlas, por esta razón, se llevó a cabo el siguiente proceso comenzando por el conjunto de datos de Documentos:

Ilustración 30

Proceso de descarte de variables no necesarias del conjunto de datos de Documentos

```
# Listado de variables por descartar
variables_por_descartar = [
    "fechaElaboracion", "fechaVencimiento",
    "totalDocumento", "informacionVehiculo",
    "informacionDespachador", "estadoProceso",
    "estadoProcesoDetalle", "estadoProcesoActividad",
    "codigoUnificado", "codigoUnificado02",
    "nombreEmpresaTercero", "nombreComercialTercero",
    "tipoIdentificacion", "documentoIdentificacion",
    "direccionPrincipal", "direccionInterseccion",
    "direccionNumero", "direccionReferencia",
    "telefono01", "telefono02",
    "telefono03", "correoElectronico",
    "nombreFacilidad", "seriePuntoVenta",
    "nombreCaja", "serieCaja",
    "id_documento", "id_despachador",
    "id_cliente", "id_direccion_cliente",
]
# Descarte de Las variables no necesarias
documentos.drop(columns=variables_por_descartar, inplace=True)
```

Por otro lado, con respecto al descarte de variables no necesarias del conjunto de datos de Vehículos, se llevó a cabo un proceso similar:

Ilustración 31

Proceso de descarte de variables no necesarias del conjunto de datos de Vehículos

```
# Listado de variables por descartar
variables_por_descartar = [
    "fechaElaboracion", "fechaVencimiento",
    "totalDocumento", "informacionVehiculo",
    "informacionDespachador", "estadoProceso",
    "estadoProcesoDetalle", "estadoProcesoActividad",
    "codigoUnificado", "codigoUnificado02",
    "nombreEmpresaTercero", "nombreComercialTercero",
    "tipoIdentificacion", "documentoIdentificacion",
    "direccionPrincipal", "direccionInterseccion",
    "direccionNumero", "direccionReferencia",
    "telefono01", "telefono02",
    "telefono03", "correoElectronico",
    "nombreFacilidad", "seriePuntoVenta",
    "nombreCaja", "serieCaja",
    "id_documento", "id_despachador",
    "id_direccion_cliente"
]
# Descarte de Las variables no necesarias
documentos.drop(columns=variables_por_descartar, inplace=True)
```

De esta forma, una vez concretados los procesos de descarte de las variables no relevantes, se confirmó que se eliminaron 30 variables del conjunto de datos de Documentos y 12 del de Vehículos, resultando en las nuevas dimensiones que pueden ser visualizadas en la siguiente ilustración:

Ilustración 32

Vista de las nuevas dimensiones de los conjuntos de datos

```

...
Vista de las nuevas dimensiones de los conjuntos de datos
...

# Conjunto de datos de Documentos
print(f"El conjunto de datos de Documentos ahora posee {documentos.shape[1]} variables.")
# Conjunto de datos de Vehículos
print(f"El conjunto de datos de Vehículos ahora posee {vehiculos.shape[1]} variables.")

El conjunto de datos de Documentos ahora posee 7 variables.
El conjunto de datos de Vehículos ahora posee 4 variables.

```

6.3. Resumen de variables que permanecerán

De este modo, habiéndose descartado todas las variables que fueron consideradas como no necesarias en el conjunto de datos de Documentos tanto como en el de Vehículos, las relevantes que serán utilizadas para el desarrollo de LUMA han sido enlistadas en las siguientes tablas comenzando por las del conjunto de datos de Documentos:

Tabla 21

VARIABLES QUE PERMANECERÁN EN EL CONJUNTO DE DATOS DE DOCUMENTOS

| Variable | Motivos |
|--------------|--|
| id_vehiculo | La variable construida durante la exploración de datos resulta relevante porque permitirá asignar vehículos a los sectores de entrega de cada documento que el modelo de agrupación en clústeres realizará, es decir, se podrá asignar un vehículo a cada documento. |
| fechaEntrega | La variable presenta utilidad puesto que es la que identificará que documentos deberán ser despachados en el día, sin embargo, es |

| | |
|-----------------------|--|
| | necesario notar que esta variable únicamente presentará mayor relevancia para el entrenamiento, ya que, cuando este en producción, solamente trabajará con datos del día considerado. |
| totalPesoDocumento | La variable es de vital importancia porque reporta el peso del documento y será necesaria para su asignación a los distintos vehículos de despacho. |
| totalVolumenDocumento | La variable es de vital importancia porque reporta el volumen del documento y será necesaria para su asignación a los distintos vehículos de despacho. |
| estadoDocumento | La variable, si bien no será útil para el entrenamiento debido a los hallazgos de la exploración de datos, será sumamente necesaria para el reporte de los documentos que han sido asignados para su despacho al sistema Practalis, así como para identificar aquellos documentos que aún no han sido despachados. |
| coordenadaLatitud | La variable es de vital importancia pues permite identificar la latitud del punto de entrega al que se deberá despachar el documento al cual corresponda. |
| coordenadaLongitud | La variable es de vital importancia pues permite identificar la longitud del punto de entrega al que se deberá despachar el documento al cual corresponda. |

Por otro lado, las variables que permanecerán en el conjunto de datos de vehículos son las ilustradas en la siguiente tabla:

Tabla 22
Variables que permanecerán en el conjunto de datos de Vehículos

| Variable | Motivos |
|-------------------|---|
| id_vehiculo | La variable construida durante la exploración de datos resulta relevante porque permitirá identificar la capacidad del peso y volumen del vehículo que se esté considerando asignar a uno de los sectores de entrega que el modelo de agrupación en clústeres identifique, es decir, servirá para mapear la información de los vehículos. |
| capacidadVehiculo | Esta variable contiene la capacidad en cuanto al peso del vehículo, por lo que, es de vital importancia para considerar cuantos documentos podrá despachar. |
| capacidadVolumen | Esta variable contiene la capacidad en cuanto al volumen del vehículo, por lo que, es de vital importancia para considerar cuantos documentos podrá despachar. |
| estadoVehiculo | Esta variable identifica cuando un vehículo se encuentra disponible para realizar el despacho de documentos, por lo que, si bien no |

presenta utilidad para el entrenamiento de LUMA, para su producción permitirá identificar los vehículos que deberán ser asignados para el despacho de documentos.

6.4. Limpieza de datos

Una vez concluidos los procesos de selección de variables, es momento de proceder con su limpieza de acuerdo con los hallazgos de la exploración de datos y su verificación de calidad, por lo que, se procederá con una breve recapitulación de los valores faltantes del conjunto de datos de Documentos, debido a que, tras la selección de datos realizada, es posible que la cantidad de tareas para su limpieza haya disminuido considerablemente.

Ilustración 33

Valores faltantes por variable del conjunto de datos de Documentos

```
# Valores faltantes por variable del conjunto de datos de Documentos
documentos.isna().sum()
fechaEntrega          0
totalPesoDocumento    0
totalVolumenDocumento 0
estadoDocumento       0
coordenadaLatitud     3
coordenadaLongitud    0
id_vehiculo           0
dtype: int64
```

De esta forma, se evidencia que, en el caso de valores faltantes, la cantidad de tareas por realizar ha disminuido a únicamente tener que gestionar los valores nulos de la variable *coordenadaLatitud*.

Por otro lado, a partir de los resultados de la evaluación de la calidad de los datos, se determinó que algunos de los documentos poseen errores en cuanto a sus coordenadas de longitud y latitud, de manera que, a continuación, se presentará un breve resumen de estos:

Ilustración 34

Verificación de documentos con coordenadas erróneas

```
'''
Verificación de documentos con coordenadas erróneas
'''
# Condiciones donde ocurre el error
condicion_longitud = documentos["coordenadaLongitud"] == 0
condicion_latitud = documentos["coordenadaLatitud"] == 0

# Evaluación
documentos_erroneos = documentos[condicion_longitud & condicion_latitud]

# Reporte de la verificación
print(f"Cantidad de documentos erróneos: {documentos_erroneos.shape[0]}")
print(f"Porcentaje de documentos erróneos: {round(documentos_erroneos.shape[0] * 100 / documentos.shape[0], 2)}%")

Cantidad de documentos erróneos: 23
Porcentaje de documentos erróneos: 1.44%
```

Así pues, dando comienzo oficial a la limpieza de datos, se procederá con el tratamiento de los valores faltantes de la variable *coordenadaLatitud*, a los que, se les realizará un proceso de eliminación.

Esto porque la cantidad de valores nulos de la variable es insignificante con un 0.18% del total de datos, además, durante la planificación de la logística de última milla que se lleva actualmente a cabo en Practalis, los documentos que no poseen su información geográfica completa son descartados para la planificación de su despacho, por lo que, se puede optar por la eliminación de los registros con dichos valores faltantes del conjunto de datos de Documentos.

Para esto, se utilizará el método *dropna()*, el cual, eliminará todos los registros con valores faltantes en el conjunto de datos de Documentos, por lo tanto, se descartarán todos los valores faltantes identificados en la variable *coordenadaLatitud*. El proceso puede ser observado a continuación:

Ilustración 35

Eliminación de valores faltantes en el conjunto de datos de Documentos y reporte de sus resultados

```
...
Eliminación de valores faltantes
...
# Se eliminan los valores faltantes
documentos.dropna(inplace=True)

# Se reportan los resultados del proceso
print(f"El conjunto de datos de Documentos ahora posee: {documentos.shape[0]} documentos.")
print("Además, ahora posee los siguientes valores faltantes por variable:")
documentos.isna().sum()
```

El conjunto de datos de Documentos ahora posee: 1599 documentos.
Además, ahora posee los siguientes valores faltantes por variable:

```
fechaEntrega          0
totalPesoDocumento    0
totalVolumenDocumento 0
estadoDocumento       0
coordenadaLatitud     0
coordenadaLongitud   0
id_vehiculo           0
dtype: int64
```

Por otra parte, ha llegado el momento de procesar los documentos cuyas coordenadas de longitud y latitud sean erróneas, para ello, similar al tratamiento realizado para los valores faltantes, se procederá con la eliminación de estos registros:

Ilustración 36

Procesamiento de documentos erróneos

```
...
Eliminación de documentos erróneos
...
# Condición donde no ocurre el error
condicion_sin_errores = (documentos["coordenadaLatitud"] != 0) | (documentos["coordenadaLongitud"] != 0)

# Descarte de los documentos erróneos
documentos = documentos[condicion_sin_errores]

# Información para el reporte de resultados del proceso
condicion_longitud = documentos["coordenadaLongitud"] == 0
condicion_latitud = documentos["coordenadaLatitud"] == 0
documentos_erroneos = documentos[condicion_longitud & condicion_latitud]

# Reporte de resultados del proceso
print(f"Cantidad de documentos erróneos: {documentos_erroneos.shape[0]}")
print(f"Porcentaje de documentos erróneos: {round(documentos_erroneos.shape[0] * 100 / documentos.shape[0], 2)}%")
```

Cantidad de documentos erróneos: 0
Porcentaje de documentos erróneos: 0.0%

Adicionalmente, asimismo con respecto al conjunto de datos de Documentos, durante la exploración de datos también se identificó la presencia de datos anormales que no serán tratados debido al alcance de LUMA, estos corresponden a aquellos documentos cuyo peso o volumen registrado es superior a la capacidad máxima del vehículo con mayor capacidad en cuanto a dichas medidas, por lo que, a continuación, se podrá visualizar la identificación de estos documentos anormales:

Ilustración 37

Identificación de documentos anormales

```
'''
Identificación de documentos anormales
'''
# Se identifica la capacidad máxima de todos los vehículos en cuanto al peso y volumen
capacidad_maxima_peso = vehiculos["capacidadVehiculo"].max()
capacidad_maxima_volumen = vehiculos["capacidadVolumen"].max()

# Se identifica la cantidad de documentos cuyo peso o volumen es superior
# al de la capacidad máxima de los vehículos
documentos_anormales_peso = documentos[documentos["totalPesoDocumento"] > capacidad_maxima_peso]
documentos_anormales_volumen = documentos[documentos["totalVolumenDocumento"] > capacidad_maxima_volumen]

# Se reporta la cantidad de documentos anormales identificados en ambos casos
print("La cantidad de documentos anormales con respecto al peso fue de:", documentos_anormales_peso.shape[0])
print("La cantidad de documentos anormales con respecto al volumen fue de:", documentos_anormales_volumen.shape[0])

La cantidad de documentos anormales con respecto al peso fue de: 20
La cantidad de documentos anormales con respecto al volumen fue de: 20
```

De esta manera, surge el cuestionamiento de si los documentos anormales identificados en cuanto al peso son los mismos que los identificados con respecto al volumen, por lo tanto, con base al código en la siguiente ilustración, se pudo determinar que, en efecto, las anomalías corresponden a los mismos registros:

Ilustración 38

Evaluación de igualdad entre los casos de anomalía presentados en los documentos

```
# Se evalua si ambas anomalías corresponden a los mismos documentos
documentos_anormales_peso == documentos_anormales_volumen
```

| | fechaEntrega | coordenadaLatitud | coordenadaLongitud | totalPesoDocumento | totalVolumenDocumento | estadoDocumento | id_vehiculo |
|------|--------------|-------------------|--------------------|--------------------|-----------------------|-----------------|-------------|
| 196 | True | True | True | True | True | True | True |
| 763 | True | True | True | True | True | True | True |
| 872 | True | True | True | True | True | True | True |
| 884 | True | True | True | True | True | True | True |
| 1010 | True | True | True | True | True | True | True |
| 1108 | True | True | True | True | True | True | True |
| 1117 | True | True | True | True | True | True | True |
| 1129 | True | True | True | True | True | True | True |
| 1194 | True | True | True | True | True | True | True |
| 1274 | True | True | True | True | True | True | True |
| 1277 | True | True | True | True | True | True | True |
| 1311 | True | True | True | True | True | True | True |
| 1352 | True | True | True | True | True | True | True |
| 1426 | True | True | True | True | True | True | True |
| 1441 | True | True | True | True | True | True | True |
| 1456 | True | True | True | True | True | True | True |
| 1506 | True | True | True | True | True | True | True |
| 1521 | True | True | True | True | True | True | True |
| 1525 | True | True | True | True | True | True | True |
| 1551 | True | True | True | True | True | True | True |

En consecuencia, el tratamiento de estas anomalías implica únicamente el tratamiento de una de las dos situaciones de anomalía, por lo que, a través del código presentado en la ilustración a continuación, se procede con la eliminación de los documentos anormales identificados:

Ilustración 39

Eliminación de documentos anormales y reporte de sus resultados

```
'''
Eliminación de documentos anormales
'''
# Se identifica la capacidad máxima de todos los vehículos en cuanto al peso
capacidad_maxima_peso = vehiculos["capacidadVehiculo"].max()

# Se identifica la condición en la que los documentos no son anormales
condicion_normalidad = documentos["totalPesoDocumento"] <= capacidad_maxima_peso

# Se eliminan los documentos anormales
documentos = documentos[condicion_normalidad]

'''
Reporte del tratamiento realizado
'''
# Se identifica la capacidad máxima de todos los vehículos en cuanto al peso y volumen
capacidad_maxima_peso = vehiculos["capacidadVehiculo"].max()
capacidad_maxima_volumen = vehiculos["capacidadVolumen"].max()

# Se identifica la cantidad de documentos cuyo peso o volumen es superior
# al de la capacidad máxima de los vehículos
documentos_anormales_peso = documentos[documentos["totalPesoDocumento"] > capacidad_maxima_peso]
documentos_anormales_volumen = documentos[documentos["totalVolumenDocumento"] > capacidad_maxima_volumen]

# Se reporta la cantidad de documentos anormales identificados en ambos casos
print("La cantidad de documentos anormales con respecto al peso fue de:", documentos_anormales_peso.shape[0])
print("La cantidad de documentos anormales con respecto al volumen fue de:", documentos_anormales_volumen.shape[0])

La cantidad de documentos anormales con respecto al peso fue de: 0
La cantidad de documentos anormales con respecto al volumen fue de: 0
```

Finalmente, con respecto al conjunto de datos de Vehículos, con base a los hallazgos de la verificación de la calidad de los datos y su exploración, se identificó la presencia de valores erróneos en la variable *anioVehiculo*, sin embargo, debido a que esta variable fue eliminada durante la selección de datos, implica que no se requiere ningún proceso de limpieza para los vehículos.

6.5. Formateo de variables

Otro aspecto por considerar será la del formateo de variables del conjunto de datos de Documentos, pues, durante su exploración se identificó que los registros de la variable *estadoDocumento* abren espacio a mucha incertidumbre, y, ya que esta variable permitirá identificar los documentos a los que se deberá planificar su despacho una vez que LUMA este en

producción, entonces, se procederá con su formateo con el fin de que todos sus datos correspondan al estado en el que se encuentren listos para la planificación de su despacho.

De esta forma, permitirá entrenar a LUMA para que considere los documentos que debe planificar y reportar como despachados.

Adicionalmente, también se realizará el proceso de formateo para la variable *id_vehiculo* del conjunto de datos de Documentos, ya que, similar a *estadoDocumento* dada la incertidumbre presentada durante la exploración de datos, esta variable únicamente funcionará para indicar que vehículos han sido asignados a que documentos y, por ende, a los sectores que el modelo de agrupación en clústeres identifique.

Por lo tanto, no tiene relevancia que la variable posea los identificadores de vehículos a los que previamente hayan sido asignados los diversos documentos.

El proceso de formateo de datos para ambas situaciones puede ser observado a continuación:

Ilustración 40

Formateo de la variable *estadoDocumento* e *id_vehiculo* del conjunto de datos de Documentos

```
...
Formateo de datos de la variable estadoDocumento
...
# Se formatean los datos para que todos los estados correspondan
# a estar listos para la planificación de su despacho.
documentos["estadoDocumento"] = 3

# Se formatean los datos para que todos los documentos no se
# encuentren asignados a ningún vehículo
documentos["id_vehiculo"] = "Sin asignar"

# Información para el reporte
estados = documentos["estadoDocumento"].unique()
vehiculos_asignados = documentos["id_vehiculo"].unique()

# Se reporta los resultados del proceso
print(f"Los estados que posee la variable estadoDocumento son: {estados}")
print(f"Los vehículos que han sido asignados a los documentos son: {vehiculos_asignados}")

Los estados que posee la variable estadoDocumento son: [3]
Los vehículos que han sido asignados a los documentos son: ['Sin asignar']
```

6.6. Estructuración de los conjuntos de datos

Como último punto se procederá a estructurar el orden de las variables de los conjuntos de datos de Documentos y Vehículos, esto, con la finalidad de hacer su visualización fácil e intuitiva, por lo que, se llevó a cabo el siguiente código para realizarlo:

Ilustración 41

Estructuración del orden de las variables de los conjuntos de datos de Documentos y Vehículos

```
...  
Estructuración del orden de las variables de los conjuntos de datos  
...  
  
# Orden del conjunto de datos de Documentos  
orden_documentos = [  
    "fechaEntrega",  
    "coordenadaLatitud",  
    "coordenadaLongitud",  
    "totalPesoDocumento",  
    "totalVolumenDocumento",  
    "estadoDocumento",  
    "id_vehiculo"  
]  
  
# Orden del conjunto de datos de Vehículos  
orden_vehiculos = [  
    "id_vehiculo",  
    "capacidadVehiculo",  
    "capacidadVolumen",  
    "estadoVehiculo"  
]  
  
# Estructuración del nuevo orden para los conjuntos de datos  
documentos = documentos[orden_documentos]  
vehiculos = vehiculos[orden_vehiculos]
```

6.7. Resultados de la preparación de datos

Durante esta etapa se prepararon los conjuntos de datos de Documentos y Vehículos para su uso en el desarrollo de LUMA, por lo que, se llevaron a cabo procesos de selección y formateo de variables, limpieza de datos faltantes y erróneos, y, finalmente, estructuración de la disposición de los conjuntos preparados.

En este sentido, en este subcapítulo se presentará a manera de resumen todos los logros alcanzados durante este proceso de preparación de datos.

Por lo que, para comenzar, a partir de la selección de variables, los conjuntos de datos fueron redimensionados a un total de 7 variables para el conjunto de datos de Documentos y a un total de 4 para el de Vehículos, además, durante la limpieza se procedió con la eliminación de varios registros del conjunto de Documentos, por ende, sus dimensiones en cuanto a registros fueron reducidas de 1602 registros a 1556, es decir, se eliminaron 46 registros de documentos.

Finalmente, se realizó un proceso de formateo y estructuración de datos que permitió una visualización más sencilla e intuitiva de los conjuntos de datos de Documentos y Vehículos.

Los conjuntos de datos resultantes son los siguientes:

Ilustración 42

Visualización del conjunto de datos de Documentos preparado

```
# Se visualiza un head de los datos de Documentos
documentos.head()
```

| | fechaEntrega | coordenadaLatitud | coordenadaLongitud | totalPesoDocumento | totalVolumenDocumento | estadoDocumento | id_vehiculo |
|---|--------------|-------------------|--------------------|--------------------|-----------------------|-----------------|-------------|
| 0 | 2024-03-11 | ████████ | ████████ | 4000.0 | 100.0 | 3 | Sin asignar |
| 1 | 2024-03-11 | ████████ | ████████ | 3050.0 | 71.0 | 3 | Sin asignar |
| 2 | 2024-03-11 | ████████ | ████████ | 9000.0 | 200.0 | 3 | Sin asignar |
| 3 | 2024-03-11 | ████████ | ████████ | 922.4 | 19.9 | 3 | Sin asignar |
| 4 | 2024-03-11 | ████████ | ████████ | 1125.0 | 22.5 | 3 | Sin asignar |

Ilustración 43

Visualización del conjunto de datos de Vehículos preparado

```
# Se visualiza un head de Los datos de Vehículos  
vehiculos.head()
```

| | id_vehiculo | capacidadVehiculo | capacidadVolumen | estadoVehiculo |
|---|-------------|-------------------|------------------|----------------|
| 0 | 111 | 10000 | 200 | A |
| 1 | 112 | 6000 | 120 | A |
| 2 | 113 | 15500 | 310 | A |
| 3 | 114 | 6750 | 135 | A |
| 4 | 115 | 15500 | 310 | A |

CAPÍTULO VII: DESARROLLO DE LUMA

7. Desarrollo de LUMA

7.1. Selección de variables que serán utilizadas por el modelo de agrupación en clústeres

El primer paso para el desarrollo de LUMA será la selección de las variables que se utilizarán para la agrupación en clústeres, por lo que, naturalmente, las seleccionadas fueron: *coordenadaLatitud* y *coordenadaLongitud*. Esto se debe a la necesidad de agrupar los puntos de entrega según sus ubicaciones geográficas, permitiendo que los vehículos puedan dirigirse a sectores geográficamente cercanos, optimizando así las rutas de entrega.

El código utilizado para esta selección de variables fue el siguiente:

Ilustración 44

Código para la selección de variables que serán utilizadas por el modelo de agrupación en clústeres

```
# Se identifican las variables que serán utilizadas para la agrupación en clústeres
variables_por_considerar = [
  "coordenadaLatitud",
  "coordenadaLongitud"
]

# Se seleccionan las variables que serán utilizadas para la agrupación en clústeres
X = documentos_aleatorios[variables_por_considerar]
```

```
# Visualización de la selección
X.head()
```

| | coordenadaLatitud | coordenadaLongitud |
|-----|-------------------|--------------------|
| 647 | ██████████ | ██████████ |
| 648 | ██████████ | ██████████ |
| 649 | ██████████ | ██████████ |
| 650 | ██████████ | ██████████ |
| 651 | ██████████ | ██████████ |

7.2. Identificación de parámetros óptimos

Por otra parte, para agrupar los documentos y determinar sus sectores de entrega, se utilizará el modelo KMeans, el cual, será fundamental para la automatización de la logística de última milla.

Sin embargo, KMeans requiere un parámetro conocido como **k**, que indica el número de clústeres que el modelo debe identificar, por lo tanto, es crucial desarrollar un código que permita determinar este parámetro utilizando la técnica del coeficiente de silueta.

Esta técnica es un método de validación utilizado para evaluar la calidad de una agrupación en clústeres, pues, este coeficiente mide qué tan similar es un objeto a su propio clúster en comparación con otros clústeres, proporcionando una medida de coherencia interna. Los valores del coeficiente de silueta oscilan entre -1 y 1, donde un valor cercano a 1 indica que los objetos están bien agrupados, un valor cercano a 0 sugiere que los objetos están en los límites de los clústeres y un valor negativo indica que los objetos podrían estar agrupados incorrectamente.

Por lo que, esta técnica es útil para determinar el número óptimo de clústeres en un conjunto de datos, no obstante, para hacerlo, es necesario definir una variable llamada **k_max**, la cual, corresponde al número máximo de clústeres que se evaluarán mediante el coeficiente de silueta con el objetivo de identificar el valor de **k** más adecuado para la agrupación de los puntos de entrega de los documentos. De manera que, para comenzar, se debe determinar el valor de **k_max**.

Para ello, idealmente, **k_max** debería ser igual al número total de vehículos disponibles, ya que los puntos de entrega no deben ser sectorizados en una cantidad que no pueda ser

manejada; no obstante, en algunos días, la cantidad de documentos por despachar es menor que la cantidad de vehículos disponibles, por esta razón, es necesario crear un algoritmo que pueda identificar el valor más adecuado para **k_max** en estas circunstancias.

A continuación, se presentará el algoritmo realizado:

Ilustración 45

Función que permite la identificación del parámetro **k_max**

```
def identificacion_kmax(documentos, vehiculos):  
    """  
    La función identifica el valor óptimo para k_max considerando el total de documentos por  
    despachar en el día y el total de vehículos disponibles para su despacho.  
    """  
    # Se prepara la variable k_max para su posterior asignación óptima  
    k_max = 0  
  
    # Se identifican la cantidad de documentos por despachar  
    total_documentos = documentos.shape[0]  
    # Se identifica la cantidad de vehículos disponibles  
    total_vehiculos = vehiculos[vehiculos["estadoVehiculo"] == "A"].shape[0]  
  
    # Se evalúa si el total de documentos es menor al total de vehículos disponibles  
    if total_documentos < total_vehiculos:  
        # Si resulta verdadero, entonces k_max será igual al total de documentos por despachar  
        k_max = total_documentos  
    else:  
        # En cambio, si resulta falso, k_max será igual al total de vehículos disponibles  
        k_max = total_vehiculos  
    # Se retorna el valor óptimo para k_max  
    return k_max
```

Una vez determinado el valor de **k_max**, el siguiente paso es proceder con la identificación del número óptimo de clústeres (**k**), por lo que, como se mencionó previamente, la metodología empleada para hallar este valor óptimo se basa en el análisis del coeficiente de silueta. A continuación, se podrá observar el código correspondiente para identificar el valor de **k** óptimo:

Ilustración 46

Función que permite la identificación del valor de **k** óptimo

```
def get_k_optima(k_max, X):  
    """  
    La función devuelve el valor óptimo de k mediante la utilización del método Silhouette.  
    """  
    # Se instancia una lista para almacenar los silhouette scores promedios para cada k  
    silhouette_avg_scores = []  
  
    # Se itera sobre la suposición de que el valor de k óptimo estará entre 2 a k_max clústeres  
    for k in range(2, k_max + 1):  
        # Se crea un modelo KMeans con la cantidad de clústeres que se esté considerando  
        kmeans = KMeans(n_clusters=k, n_init="auto", random_state=42)  
        # Se ajusta el modelo y se predicen las etiquetas de los clústeres  
        etiquetas = kmeans.fit_predict(X)  
        # Se calcula el silhouette score para las etiquetas de los clústeres  
        silhouette_avg = silhouette_score(X, etiquetas)  
        # Se agrega el silhouette score a la lista previamente definida  
        silhouette_avg_scores.append(silhouette_avg)  
  
    # Se obtiene el valor máximo (más cercano a 1) de los silhouette scores  
    max_silhouette_score = max(silhouette_avg_scores)  
  
    # Se retorna el índice del silhouette score más alto, ajustado por el rango de k más 1  
    return silhouette_avg_scores.index(max_silhouette_score) + 2
```

7.3. Agrupación en clústeres mediante KMeans

Una vez identificado el número óptimo de clústeres mediante el coeficiente de silueta, el siguiente paso es implementar la agrupación en clústeres utilizando el modelo KMeans, de manera que, en esta etapa es fundamental configurar el modelo para que detecte el número de clústeres igual al valor óptimo de **k**, y, además, es crucial fijar el *random_state* al mismo valor que fue utilizado para la identificación de dicho valor, pues, permitirá asegurar una agrupación acorde a los resultados de los coeficientes de silueta.

A continuación, se podrá observar el código realizado para dicho propósito y, adicionalmente, como se debería entrenar y agregar los sectores identificados a un dataframe con documentos por despachar, ya que, es fundamental para la planificación de sus entregas:

Ilustración 47

Agrupación en clústeres, modelo, entrenamiento y asignación

```
...  
Se realiza la agrupación en clústeres mediante KMeans y se asignan los documentos  
a los sectores identificados  
...  
# Se crea un modelo kmeans dónde se identificarán clústers igual a k_optimo  
kmeans = KMeans(n_clusters=k_optimo, random_state=42, n_init="auto")  
  
# Se ajusta el modelo y se predicen las etiquetas de los clústeres  
etiquetas = kmeans.fit_predict(X)  
  
# Se agregan las etiquetas predecidas a los documentos por despachar  
documentos_aleatorios["sector"] = etiquetas
```

7.4. Algoritmos de asignación de vehículos a los sectores identificados y posterior asignación de documentos a dichos vehículos

Habiendo identificado los sectores de entrega de los documentos que se deben despachar en el día, es necesario realizar un proceso de asignación de vehículos a cada uno de estos sectores y, posteriormente, asignar cada documento a dichos vehículos.

Por lo tanto, considerando que el objetivo del negocio es reducir el porcentaje de subutilización de los vehículos, la premisa debe centrarse en minimizar esta subutilización, de manera que, el algoritmo se enfocará inicialmente en los sectores con la menor cantidad de documentos, peso y volumen para la planificación de su logística de última milla, ya que, así se puede asegurar de mejor manera el despacho de todos los documentos asignados a cada sector.

El primer paso para lograr esto es crear un dataframe que contenga la información de cada sector:

Ilustración 48

Creación de un dataframe con información de los sectores identificados

```
'''
Se crea un dataframe que contenga los datos de cada sector.
'''
# Se instancia una lista que será utilizado para la creación del nuevo dataframe
datos_sectores = []

# Se itera sobre todos los sectores identificados en orden
for sector in sorted(documentos_aleatorios["sector"].unique().tolist()):
    # Se crea un diccionario que será utilizado para almacenar los datos del sector
    datos_sector = {
        "sector" : sector,
        "total_documentos" : total_documentos,
        "peso_total" : peso_total_sector,
        "volumen_total" : volumen_total_sector
    }
    # Se agrega el diccionario a la lista de los datos con los que se creará el nuevo dataframe
    datos_sectores.append(datos_sector)

# Se crea el nuevo dataframe con los datos de los sectores
sectores = pd.DataFrame(datos_sectores)
```

Así pues, ahora ya es factible proceder con la realización de los algoritmos de asignación de vehículos a los sectores identificados y posterior asignación de documentos a dichos vehículos, el código completo puede ser visualizado en las ilustraciones a continuación:

Ilustración 49

Ilustración 1 del algoritmo de asignación de vehículos a los sectores identificados y posterior asignación de documentos a dichos vehículos

```
'''
Algoritmo de asignación de vehículos a sectores y de documentos de cada sector a dichos vehículos.
'''
# Se ordenan los datos del nuevo dataframe de acuerdo a la cantidad de documentos que se deberán despachar
sectores = sectores.sort_values(by="total_documentos", ascending=True)

# Se ordenan los vehículos de acuerdo a su capacidad respecto al peso
vehiculos = vehiculos.sort_values(by="capacidadVehiculo", ascending=False)
vehiculos = vehiculos.reset_index(drop=True)

# Se iteran sobre las filas del dataframe ordenado
for index_sec, row_sec in sectores.iterrows():

    # Se seleccionan los vehículos que están disponibles para realizar despachos
    vehiculos_disponibles = vehiculos[vehiculos["estadoVehiculo"] == "A"]

    # Se instancia una lista que almacenará todos los vehículos asignados al sector
    vehiculos_asignados = []

    # Se identifica el peso total que se deberá despachar en el sector
    peso_restante = row_sec["peso_total"]

    # Si no existen vehículos disponibles, no se realizarán más asignaciones
    if vehiculos_disponibles.shape[0] == 0:
        break
```

Ilustración 50

Ilustración 2 del algoritmo de asignación de vehículos a los sectores identificados y posterior asignación de documentos a dichos vehículos

```
...
Parte 1: Algoritmo de asignación inicial de vehículos a los sectores identificados
...
while True:
    # Se identifica la subutilización que poseerían los vehículos si se les asignará al sector
    subutilizaciones = []
    for index_veh, row_veh in vehiculos_disponibles.iterrows():
        subutilizacion = (row_veh["capacidadVehiculo"] - peso_restante) * 100 / row_veh["capacidadVehiculo"]
        subutilizaciones.append((row_veh["id_vehiculo"], subutilizacion))
    # Se ordena la lista de acuerdo al porcentaje de subutilización
    subutilizaciones.sort(key=lambda x: x[-1], reverse=True)

    # Se evalúa si ha existido una posterior asignación de vehículos al sector
    if len(vehiculos_asignados) != 0:
        # En cuyo caso, entonces se omiten los vehículos ya asignados
        subutilizaciones = [veh for veh in subutilizaciones if veh[0] not in vehiculos_asignados]

    # Se asigna el vehículo con menor subutilización reportada al sector
    vehiculos_asignados.append(subutilizaciones.pop()[0])

    # Se evalúa el peso total que el o los vehículos asignados son capaces de despachar
    capacidad_acumulada = 0
    for veh_asignado in vehiculos_asignados:
        capacidad_acumulada += vehiculos[vehiculos["id_vehiculo"] == veh_asignado]["capacidadVehiculo"].iloc[0]
    peso_restante = row_sec["peso_total"] - capacidad_acumulada

    # Si el peso restante es menor o igual a 0 se finaliza la asignación de vehículos
    if peso_restante <= 0:
        break
```

Ilustración 51

Ilustración 3 del algoritmo de asignación de vehículos a los sectores identificados y posterior asignación de documentos a dichos vehículos

```
...
Parte 2: Algoritmo de verificación de vehículos asignados con respecto al peso de cada documento
...
# Se identifican los pesos de todos los documentos
pesos = documentos_aleatorios[documentos_aleatorios["sector"] == row_sec["sector"]]["totalPesoDocumento"].tolist()
# Se ordenan los pesos de menor a mayor
pesos.sort()
# Se define una variable que identifica si hubo algún cambio en los vehículos asignados
hubo_cambio = True
# Se define una lista que almacenará todos los pesos de los documentos que generen problema
pesos_problematicos = []

while hubo_cambio:
    # Se tiene la hipótesis de que no habrá cambios en las asignaciones de los vehículos
    hubo_cambio = False
    # Se evalúan los vehículos asignados desde el que posee más capacidad
    for id_vehiculo in vehiculos_asignados[::-1]:
        for peso in pesos:
            # Si el peso de un documento sobrepasa la capacidad del vehículo considerado...
            if peso > vehiculos[vehiculos["id_vehiculo"] == id_vehiculo]["capacidadVehiculo"].iloc[0]:
                # Si el peso considerado no fue considerado antes, entonces se lo procesa
                if pesos.index(peso) not in pesos_problematicos:
                    # Se guarda la información del peso que generó el evento
                    pesos_problematicos.append(pesos.index(peso))

            # Se identifica el primer vehículo que sea capaz de llevar el peso del documento
            vehiculos_adequados = vehiculos_disponibles[vehiculos_disponibles["capacidadVehiculo"] > peso]
            id_nuevo_vehiculo = vehiculos_adequados["id_vehiculo"].iloc[-1]
```

Ilustración 52

Ilustración 4 del algoritmo de asignación de vehículos a los sectores identificados y posterior asignación de documentos a dichos vehículos

```
# Se reporta el peso problemático como -1 para futuras evaluaciones de posibles de pesos iguales
pesos[pesos.index(peso)] = -1

# Se descarta el vehículo que fue incapaz
vehiculos_asignados.pop()

# Se asigna el nuevo vehículo
vehiculos_asignados.append(id_nuevo_vehiculo)

# Se reporta que hubo un cambio en los datos
hubo_cambio = True
# Si se reporto algún cambio, se reinicia el bucle de identificación
if hubo_cambio:
    hubo_cambio = False
    break

...
Parte 3: Algoritmo de evaluación de necesidad de vehículos asignados
...
# Se instancia una variable que almacenará los vehículos evaluados
vehiculos_evaluados = []

while True:
    # Se evalúa si con la nueva asignación, se podría descartar el uso de algún vehículo
    subutilizaciones = []

    # Se identifica el total de la capacidad en cuanto a peso de los vehículos evaluados
    capacidad_total = 0
    for vehiculo_eval in vehiculos_evaluados:
        capacidad_total += vehiculos[vehiculos["id_vehiculo"] == vehiculo_eval]["capacidadVehiculo"].iloc[0]

    # Se vuelve a identificar el peso total por despachar menos la capacidad de los vehículos evaluados
    peso_total = row_sec["peso_total"] - capacidad_total
```

Ilustración 53

Ilustración 5 del algoritmo de asignación de vehículos a los sectores identificados y posterior asignación de documentos a dichos vehículos

```
# Se obtienen los porcentajes de subutilización de todos los vehículos que se están considerando
for veh_asig, veh_asig in vehiculos[vehiculos["id_vehiculo"].isin(vehiculos_asignados)].iterrows():
    # Menos de los vehículos que ya han sido evaluados como asignados
    if veh_asig["id_vehiculo"] in vehiculos_evaluados:
        continue
    sub = (veh_asig["capacidadVehiculo"] - peso_total) * 100 / veh_asig["capacidadVehiculo"]
    subutilizaciones.append((veh_asig["id_vehiculo"], sub))

# Se evalúa si existen subutilizaciones positivas
subutilizaciones_positivas = [x for x in subutilizaciones if x[1] >= 0]
# Si existen subutilizaciones positivas, se asigna únicamente el vehículo con subutilización más cercana a 0
if len(subutilizaciones_positivas) > 0:
    vehiculos_evaluados.append(min(subutilizaciones_positivas, key=lambda x: abs(x[1]))[0])
    break
# Si todas las subutilizaciones son negativas, entonces, se procede con la asignación del vehículo
# con un porcentaje más cercano a 0
else:
    # Se selecciona el vehículo con el porcentaje de subutilización más cercano a 0
    vehiculo_menor_subutilizacion = min(subutilizaciones, key=lambda x: abs(x[1]))[0]
    # Se asigna el vehículo a los vehículos evaluados
    vehiculos_evaluados.append(vehiculo_menor_subutilizacion)

# Se asignan los vehículos evaluados a los vehículos asignados
vehiculos_asignados = vehiculos_evaluados

...
Parte 4: Algoritmo de asignación de documentos a los vehículos asignados al sector
...
# Se identifican los documentos que han sido asignados al sector
documentos_del_sector = documentos_aleatorios[documentos_aleatorios["sector"] == row_sec["sector"]].copy()
# Se ordenan los documentos de acuerdo al peso
documentos_del_sector = documentos_del_sector.sort_values(by="totalPesoDocumento", ascending=False)
```

Ilustración 54

Ilustración 6 del algoritmo de asignación de vehículos a los sectores identificados y posterior asignación de documentos a dichos vehículos

```
# Se identifican Los vehículos que han sido asignados al sector
vehiculos_del_sector = vehiculos[vehiculos["id_vehiculo"].isin(vehiculos_asignados)].copy()
# Se ordenan Los vehículos de acuerdo con su capacidad respecto al peso
vehiculos_del_sector = vehiculos_del_sector.sort_values(by="capacidadVehiculo", ascending=False)

# Se inicia con La asignación de documentos del sector a Los vehículos asignados al mismo
for index_veh_sec, row_veh_sec in vehiculos_del_sector.iterrows():
    # Se identifican Las capacidades del vehículo considerado
    capacidad_restante_peso = row_veh_sec["capacidadVehiculo"]
    capacidad_restantes_volumen = row_veh_sec["capacidadVolumen"]

    # Se asigna el vehículo a Los documentos que se Le sean posibles
    for index_doc_sec, row_doc_sec in documentos_del_sector.iterrows():
        # Se identifica el peso y volumen del documento considerado
        peso_documento = row_doc_sec["totalPesoDocumento"]
        volumen_documento = row_doc_sec["totalVolumenDocumento"]

        # Se evalua si al documento aun no se Le ha asignado algún vehículo
        if row_doc_sec["id_vehiculo"] == "Sin asignar":
            # Se evalua si el vehículo puede despacharlo
            if (peso_documento <= capacidad_restante_peso) and (volumen_documento <= capacidad_restantes_volumen):
                # Se actualiza La información de las capacidades del vehículo
                capacidad_restante_peso -= peso_documento
                capacidad_restantes_volumen -= volumen_documento

                # Se asigna el vehículo al documento considerado
                documentos_del_sector.at[index_doc_sec, "id_vehiculo"] = row_veh_sec["id_vehiculo"]

        # Se reporta el nuevo estado del documento
        documentos_del_sector.at[index_doc_sec, "estadoDocumento"] = 4
```

Ilustración 55

Ilustración 7 del algoritmo de asignación de vehículos a los sectores identificados y posterior asignación de documentos a dichos vehículos

```
...
Parte 5: Integración de asignaciones con el dataframe de documentos por despachar en el día completo
...
documentos_aleatorios[documentos_aleatorios["sector"] == row_sec["sector"]] = documentos_del_sector
...
Parte 6: Asignación de estado de los vehículos que han sido asignados
...
vehiculos.loc[vehiculos["id_vehiculo"].isin(documentos_aleatorios["id_vehiculo"].unique()), "estadoVehiculo"] = "N"
```

El código de elaboración propia presentado está compuesto por 4 algoritmos principales que corresponden a partes y pasos fundamentales en la asignación de vehículos a los sectores identificados y posterior asignación de documentos a dichos vehículos. En los párrafos a continuación se explicarán los diferentes algoritmos realizados acompañados de una ilustración de su código.

Para empezar, partiendo con el algoritmo correspondiente a la parte 1, este tiene el objetivo de realizar una asignación inicial de vehículos a los sectores identificados a través de una evaluación de las subutilizaciones de cada vehículo disponible con respecto al peso total registrado en el sector.

Es decir, primero se identifica la subutilización que poseerían los vehículos si se le asignará al sector, para que, posteriormente, se evalúe si ha existido alguna asignación de vehículos previa que, en cuyo caso, se omitirían de la evaluación; de esta forma, habiendo descartado los vehículos asignados previamente, se procede a asignar el vehículo con menor porcentaje de subutilización al sector.

Inmediatamente después, se evalúa el peso total que el o los vehículos asignados hasta el momento son capaces de despachar y si son suficientes para despachar todo el peso reportado en el sector considerado, y, en caso de que los vehículos asignados sean suficientes para despachar todo el peso reportado en el sector, entonces, se continua con el algoritmo de la parte 2, si no, se vuelve a iterar.

Ilustración 56

Algoritmo de asignación inicial de vehículos a los sectores identificados

```
...
Parte 1: Algoritmo de asignación inicial de vehículos a los sectores identificados
...
while True:
    # Se identifica la subutilización que poseerían los vehículos si se les asignará al sector
    subutilizaciones = []
    for index_veh, row_veh in vehiculos_disponibles.iterrows():
        subutilizacion = (row_veh["capacidadVehiculo"] - peso_restante) * 100 / row_veh["capacidadVehiculo"]
        subutilizaciones.append((row_veh["id_vehiculo"], subutilizacion))
    # Se ordena la lista de acuerdo al porcentaje de subutilización
    subutilizaciones.sort(key=lambda x: x[-1], reverse=True)

    # Se evalua si ha existido una posterior asignación de vehículos al sector
    if len(vehiculos_asignados) != 0:
        # En cuyo caso, entonces se omiten los vehículos ya asignados
        subutilizaciones = [veh for veh in subutilizaciones if veh[0] not in vehiculos_asignados]

    # Se asigna el vehículo con menor subutilización reportada al sector
    vehiculos_asignados.append(subutilizaciones.pop()[0])

    # Se evalua el peso total que el o los vehículos asignados son capaces de despachar
    capacidad_acumulada = 0
    for veh_asignado in vehiculos_asignados:
        capacidad_acumulada += vehiculos[vehiculos["id_vehiculo"] == veh_asignado]["capacidadVehiculo"].iloc[0]
    peso_restante = row_sec["peso_total"] - capacidad_acumulada

    # Si el peso restante es menor o igual a 0 se finaliza la asignación de vehículos
    if peso_restante <= 0:
        break
```

El algoritmo correspondiente a la parte 2 tiene el objetivo de verificar a los vehículos asignados hasta el momento con respecto al peso de cada documento del sector, por lo que, comienza identificando los pesos individuales de cada documento y habilitando un bucle while que iterará mientras exista un cambio en los vehículos asignados que el algoritmo pueda producir.

De modo que, dentro del while, se evaluarán todos los vehículos asignados con respecto a los documentos, donde, si el peso de un documento sobrepasa la capacidad del vehículo que se esté evaluando, entonces, se guardará la información del peso que genera el evento y se identificará el primer vehículo capaz de llevar el peso del documento problemático.

Posteriormente, se reportará el peso problemático como -1 para facilitar futuras evaluaciones de posibles pesos iguales, se descartará el vehículo que fue incapaz de despachar el documento y se asignará el nuevo vehículo identificado.

Si durante una iteración no se identificaron más eventos donde un vehículo no sea capaz de despachar un documento, entonces, se continua con el algoritmo de la parte 3.

Ilustración 57

Ilustración 1 del algoritmo de verificación de vehículos asignados con respecto al peso de cada documento

```
...
Parte 2: Algoritmo de verificación de vehículos asignados con respecto al peso de cada documento
...
# Se identifican los pesos de todos los documentos
pesos = documentos_aleatorios[documentos_aleatorios["sector"] == row_sec["sector"]]["totalPesoDocumento"].tolist()
# Se ordenan los pesos de menor a mayor
pesos.sort()
# Se define una variable que identifica si hubo algún cambio en los vehículos asignados
hubo_cambio = True
# Se define una lista que almacenará todos los pesos de los documentos que generen problema
pesos_problematicos = []

while hubo_cambio:
    # Se tiene la hipótesis de que no habrá cambios en las asignaciones de los vehículos
    hubo_cambio = False
    # Se evalúan los vehículos asignados desde el que posee más capacidad
    for id_vehiculo in vehiculos_asignados[::-1]:
        for peso in pesos:
            # Si el peso de un documento sobrepasa la capacidad del vehículo considerado...
            if peso > vehiculos[vehiculos["id_vehiculo"] == id_vehiculo]["capacidadVehiculo"].iloc[0]:
                # Si el peso considerado no fue considerado antes, entonces se lo procesa
                if pesos.index(peso) not in pesos_problematicos:
                    # Se guarda la información del peso que genero el evento
                    pesos_problematicos.append(pesos.index(peso))

            # Se asigna el primer vehículo que sea capaz de llevar el peso del documento
            vehiculos_adecuados = vehiculos_disponibles[vehiculos_disponibles["capacidadVehiculo"] > peso]
            id_nuevo_vehiculo = vehiculos_adecuados["id_vehiculo"].iloc[-1]
```

Ilustración 58

Ilustración 2 del algoritmo de verificación de vehículos asignados con respecto al peso de cada documento

```
# Se identifica el primer vehículo que sea capaz de llevar el peso del documento
vehiculos_adecuados = vehiculos_disponibles[vehiculos_disponibles["capacidadVehiculo"] > peso]
id_nuevo_vehiculo = vehiculos_adecuados["id_vehiculo"].iloc[-1]

# Se reporta el peso problemático como -1 para futuras evaluaciones de posibles pesos iguales
pesos[pesos.index(peso)] = -1

# Se descarta el vehículo que fue incapaz
vehiculos_asignados.pop()

# Se asigna el nuevo vehículo
vehiculos_asignados.append(id_nuevo_vehiculo)

# Se reporta que hubo un cambio en los datos
hubo_cambio = True

# Si se reporta algún cambio, se reinicia el bucle de identificación
if hubo_cambio:
    hubo_cambio = False
    break
```

El algoritmo de la parte 3 tiene el objetivo de evaluar la necesidad de los vehículos que han sido asignados hasta el momento, debido a que tras el algoritmo de la parte 2, se adicionó un nuevo vehículo que pudo haber modificado la necesidad de uno u otro vehículo.

Por lo tanto, el algoritmo comienza creando una lista de vehículos evaluados, que, si bien al comienzo no tendrá contenido, a medida que se ejecute el algoritmo se ira poblando con valores, pues, de hecho, durante la primera ejecución del algoritmo, cuando se identifique la capacidad total que disponen los vehículos evaluados, su valor será de cero; no obstante, este valor ira aumentando progresivamente una vez se confirme la necesidad de los vehículos considerados hasta el momento.

Posteriormente, con base al cálculo de la capacidad actual de los vehículos evaluados, el algoritmo evaluará cuanto peso resta por despachar, con el que, se evaluarán los porcentajes de subutilización de todos los vehículos asignados que aún no hayan sido evaluados.

Finalmente, se evaluará si existen subutilizaciones positivas, que, de haberlas, se asignará el vehículo con porcentaje de subutilización más cercano a cero a los vehículos evaluados y se dará fin a la parte 3, sin embargo, si todas las subutilizaciones son negativas, entonces, se procederá con la asignación del vehículo con un porcentaje de subutilización más cercano a cero a los vehículos evaluados y se volverá a iterar.

Ilustración 59

Ilustración 1 del algoritmo de evaluación de necesidad de vehículos asignados

```
...
Parte 3: Algoritmo de evaluación de necesidad de vehículos asignados
...
# Se instancia una variable que almacenará Los vehículos evaluados
vehiculos_evaluados = []

while True:
    # Se evalua si con la nueva asignación, se podría descartar el uso de algún vehículo
    subutilizaciones = []

    # Se identifica el total de la capacidad en cuanto a peso de Los vehículos evaluados
    capacidad_total = 0
    for vehiculo_eval in vehiculos_evaluados:
        capacidad_total += vehiculos[vehiculos["id_vehiculo"] == vehiculo_eval]["capacidadVehiculo"].iloc[0]

    # Se vuelve a identificar el peso total por despachar menos la capacidad de Los vehículos evaluados
    peso_total = row_sec["peso_total"] - capacidad_total

    # Se obtienen Los porcentajes de subutilización de todos los vehículos que se están considerando
    for veh_asig, veh_asig in vehiculos[vehiculos["id_vehiculo"].isin(vehiculos_asignados)].iterrows():
        # Menos de Los vehículos que ya han sido evaluados como asignados
        if veh_asig["id_vehiculo"] in vehiculos_evaluados:
            continue
        sub = (veh_asig["capacidadVehiculo"] - peso_total) * 100 / veh_asig["capacidadVehiculo"]
        subutilizaciones.append((veh_asig["id_vehiculo"], sub))
```

Ilustración 60

Ilustración 2 del algoritmo de evaluación de necesidad de vehículos asignados

```
# Se evalua si existen subutilizaciones positivas
subutilizaciones_positivas = [x for x in subutilizaciones if x[1] >= 0]
# Si existen subutilizaciones positivas, se asigna únicamente el vehículo con subutilización más cercana a 0
if len(subutilizaciones_positivas) > 0:
    vehiculos_evaluados.append(min(subutilizaciones_positivas, key=lambda x: abs(x[1]))[0])
    break
# Si todas las subutilizaciones son negativas, entonces, se procede con la asignación del vehículo
# con un porcentaje más cercano a 0
else:
    # Se selecciona el vehículo con el porcentaje de subutilización más cercano a 0
    vehiculo_menor_subutilizacion = min(subutilizaciones, key=lambda x: abs(x[1]))[0]
    # Se asigna el vehículo a los vehículos evaluados
    vehiculos_evaluados.append(vehiculo_menor_subutilizacion)

# Se asignan Los vehículos evaluados a los vehículos asignados
vehiculos_asignados = vehiculos_evaluados
```

Finalmente, el algoritmo de la parte 4 tiene el objetivo de asignar los documentos a los vehículos que fueron asignados al sector, de manera que, el algoritmo iterará sobre todos los vehículos comenzando por los de mayor capacidad y se le asignará todos los documentos, comenzando por el más pesado, que el vehículo sea capaz de despachar; si el vehículo es incapaz de llevar más documentos, entonces, si corresponde, se procede con la asignación de documentos al vehículo siguiente.

Una vez que el o los vehículos hayan sido asignados con todos los documentos que les sean posibles despachar, entonces, se procede con el reporte del estado de los documentos como empacados y se reporta que el vehículo está a cargo del despacho de cada uno.

Ilustración 61

Algoritmo de asignación de documentos a los vehículos asignados al sector

```
...
Parte 4: Algoritmo de asignación de documentos a los vehículos asignados al sector
...
# Se identifican los documentos que han sido asignados al sector
documentos_del_sector = documentos_aleatorios[documentos_aleatorios["sector"] == row_sec["sector"]].copy()
# Se ordenan los documentos de acuerdo al peso
documentos_del_sector = documentos_del_sector.sort_values(by="totalPesoDocumento", ascending=False)

# Se identifican los vehículos que han sido asignados al sector
vehiculos_del_sector = vehiculos[vehiculos["id_vehiculo"].isin(vehiculos_asignados)].copy()
# Se ordenan los vehículos de acuerdo con su capacidad respecto al peso
vehiculos_del_sector = vehiculos_del_sector.sort_values(by="capacidadVehiculo", ascending=False)

# Se inicia con la asignación de documentos del sector a los vehículos asignados al mismo
for index_veh_sec, row_veh_sec in vehiculos_del_sector.iterrows():
    # Se identifican las capacidades del vehículo considerado
    capacidad_restante_peso = row_veh_sec["capacidadVehiculo"]
    capacidad_restantes_volumen = row_veh_sec["capacidadVolumen"]

    # Se asigna el vehículo a los documentos que se le sean posibles
    for index_doc_sec, row_doc_sec in documentos_del_sector.iterrows():
        # Se identifica el peso y volumen del documento considerado
        peso_documento = row_doc_sec["totalPesoDocumento"]
        volumen_documento = row_doc_sec["totalVolumenDocumento"]

        # Se evalúa si al documento aun no se le ha asignado algún vehículo
        if row_doc_sec["id_vehiculo"] == "Sin asignar":
            # Se evalúa si el vehículo puede despacharlo
            if (peso_documento <= capacidad_restante_peso) and (volumen_documento <= capacidad_restantes_volumen):
                # Se actualiza la información de las capacidades del vehículo
                capacidad_restante_peso -= peso_documento
                capacidad_restantes_volumen -= volumen_documento

                # Se asigna el vehículo al documento considerado
                documentos_del_sector.at[index_doc_sec, "id_vehiculo"] = row_veh_sec["id_vehiculo"]

            # Se reporta el nuevo estado del documento
            documentos_del_sector.at[index_doc_sec, "estadoDocumento"] = 4
```

El resto de código, aunque es igualmente crucial, se ha considerado que los comentarios son más que suficientes para explicar su naturaleza y relevancia para la automatización de la logística de última milla.

CAPÍTULO VIII: EVALUACIÓN

8. Evaluación

Actualmente se tienen registros de documentos emitidos para su despacho de una gran cantidad de fechas, sin embargo, si bien LUMA está diseñada para planificar el despacho de un número aleatorio de documentos, la naturaleza de su funcionamiento consiste en realizar la logística de última milla para todos los documentos que hayan sido emitidos en un solo día; en otras palabras, en producción, a LUMA se le entregará un conjunto de datos con una serie de documentos que corresponden a un sola fecha.

Por lo tanto, como primer paso, se procederá con la creación de un código que permita seleccionar un grupo de documentos a partir de fechas aleatorias, con la finalidad, de recrear el funcionamiento práctico que LUMA poseerá una vez se encuentre en producción. La función puede ser apreciada a continuación:

Ilustración 62

Función que permite obtener los documentos emitidos en una fecha aleatoria

```
def get_documentos_aleatorios():  
    """  
    La función permite obtener los documentos que fueron emitidos para su despacho en una fecha aleatoria.  
    """  
    # Se enlistan todas las fechas  
    fechas = documentos["fechaEntrega"].unique().tolist()  
  
    # Se prepara una condición para la selección de documentos de una fecha aleatoria  
    condicion = documentos["fechaEntrega"] == fechas[random.randint(0, len(documentos["fechaEntrega"].unique())-1)]  
  
    # Se seleccionan los documentos de la fecha aleatoria considerada  
    documentos_retorno = documentos[condicion]  
  
    return documentos_retorno.copy()
```

De manera que, una vez creada la función, se procedió a obtener un conjunto de documentos aleatorio, el cual, correspondió al 22 de abril del 2024, fecha en la que se emitieron 67 documentos que se encuentran distribuidos geográficamente de acuerdo con la ilustración 64.

Ilustración 63

Obtención de los documentos emitidos en una fecha aleatoria

```
# Se obtiene Los documentos emitidos de una fecha aleatoria
documentos_aleatorios = get_documentos_aleatorios()

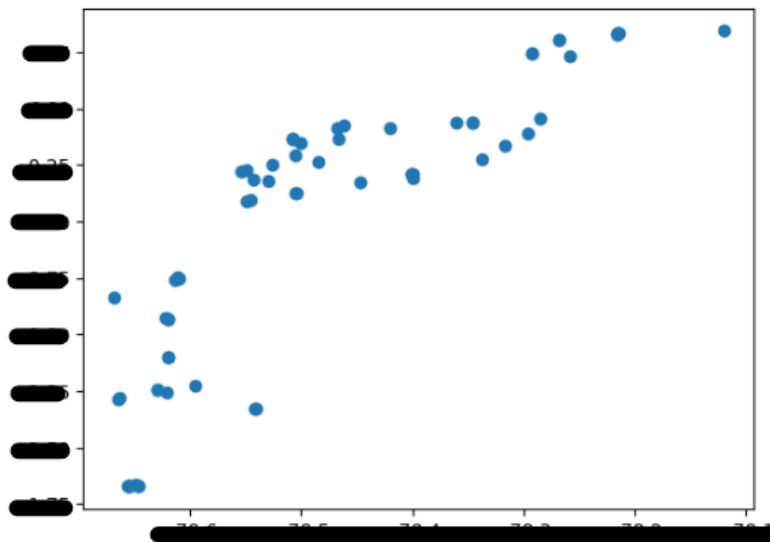
...
Reporte de información de los documentos seleccionados aleatoriamente
...
# Información para el reporte
fecha = documentos_aleatorios["fechaEntrega"].unique()[0]

print("Se ha seleccionado aleatoriamente los documentos de la fecha:", fecha)
print(f"Por lo tanto, se debe planificar el despacho de {documentos_aleatorios.shape[0]} documentos.")
```

Se ha seleccionado aleatoriamente los documentos de la fecha: 2024-03-22 00:00:00
Por lo tanto, se debe planificar el despacho de 67 documentos.

Ilustración 64

Distribución de los puntos de entrega de los documentos emitidos en una fecha aleatoria



Una vez obtenido los documentos con los que LUMA será evaluada, entonces, se procede con la selección de variables, obteniendo el siguiente dataframe:

Ilustración 65

Visualización del dataframe de las variables seleccionadas para la agrupación en clústeres

```
# Visualización de La selección
X.head()
```

| | coordenadaLatitud | coordenadaLongitud |
|-----|-------------------|--------------------|
| 647 | ██████████ | ██████████ |
| 648 | ██████████ | ██████████ |
| 649 | ██████████ | ██████████ |
| 650 | ██████████ | ██████████ |
| 651 | ██████████ | ██████████ |

Así pues, es momento de la agrupación en clústeres de los documentos que se han obtenido a través de una fecha aleatoria, de manera que, en primer lugar, se deberá identificar el valor de **k_max**:

Ilustración 66

Identificación de **k_max**

```
# Se identifica k_max
k_max = identificacion_kmax(documentos_aleatorios, vehiculos)

# Se reporta el valor de k_max
print("El valor óptimo de kmax fue de:", k_max)
```

El valor óptimo de kmax fue de: 9

Por tal motivo, habiendo obtenido el valor óptimo **k_max**, es momento de identificar el valor óptimo de **k** para los documentos en consideración:

Ilustración 67

Identificación del valor de k óptimo

```
# Se identifica el valor de k óptimo  
k_optimo = get_k_optima(k_max, X)  
  
# Se reporta el valor de k óptimo  
print("El valor de k óptimo es de:", k_optimo)
```

El valor de k óptimo es de: 5

En consecuencia, se puede observar que el total de clústeres óptimo es de 5, de modo que, a continuación, se procederá a agrupar en clústeres los documentos emitidos en la fecha aleatoria de acuerdo con sus coordenadas geográficas:

Ilustración 68

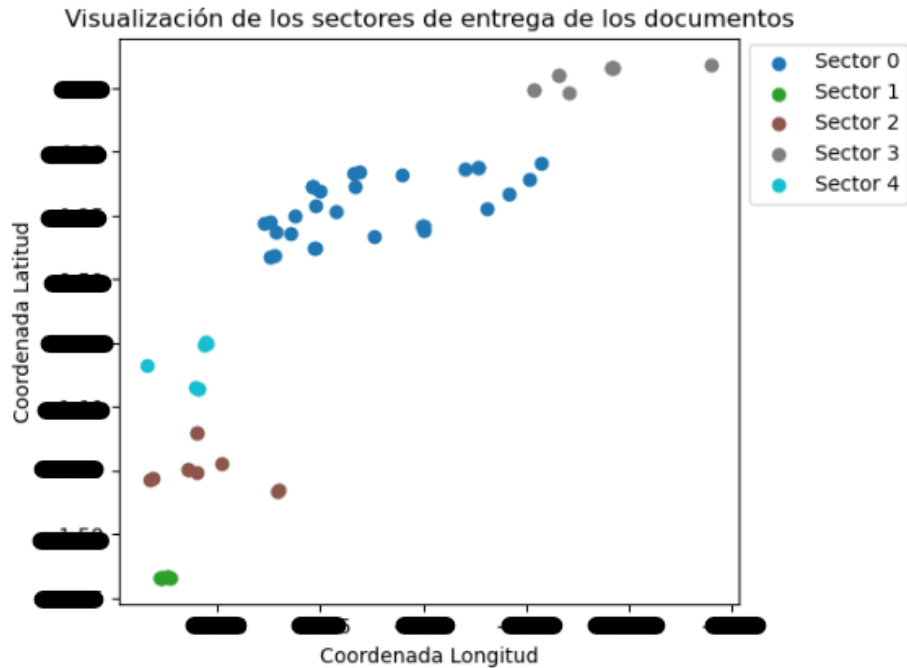
Agrupación en clústeres de los documentos emitidos en una fecha aleatoria

```
...  
Se realiza la agrupación en clústeres mediante KMeans y se asignan los documentos  
a los sectores identificados  
...  
# Se crea un modelo kmeans dónde se identificarán clústers igual a k_optimo  
kmeans = KMeans(n_clusters=k_optimo, random_state=42, n_init="auto")  
  
# Se ajusta el modelo y se predicen las etiquetas de los clústeres  
etiquetas = kmeans.fit_predict(X)  
  
# Se agregan las etiquetas prededidas a los documentos por despachar  
documentos_aleatorios["sector"] = etiquetas
```

Tras la agrupación en clústeres realizada, los sectores de entrega identificados pueden ser observados a continuación:

Ilustración 69

Visualización de los sectores de entrega de los documentos



Como es visible, se identificaron los 5 sectores que fueron calculados como óptimos a través de la técnica del coeficiente de silueta, por lo que, a continuación, se realizará un breve resumen de estos sectores y también del total de documentos que se deberá despachar:

Ilustración 70

Contextualización del día

```
...
Contextualización del día
...
print("El total de documentos por despachar en el día es de:", documentos_aleatorios.shape[0])
print("El peso total por despacharse en el día es de:", documentos_aleatorios["totalPesoDocumento"].sum())
print("El volumen total por despacharse en el día es de:", round(documentos_aleatorios["totalVolumenDocumento"].sum(), 2))

El total de documentos por despachar en el día es de: 67
El peso total por despacharse en el día es de: 114267.9
El volumen total por despacharse en el día es de: 2499.55
```

Por consiguiente, se puede intuir que los vehículos no serán capaces de despachar todos los documentos que han sido emitidos para su despacho en el día, de manera que, LUMA planificará el despacho de todos los documentos que le sean factibles realizar a los vehículos, pero, para los documentos restantes, se requerirá una nueva planificación automática.

Ilustración 71

Visualización de los resultados de la agrupación en clústeres y creación de un nuevo dataframe que contenga los datos de cada sector

```
'''
Visualización de los resultados de la agrupación en clústeres y creación de un nuevo
dataframe que contenga los datos de cada sector.
'''
# Se instancia una lista que será utilizado para la creación del nuevo dataframe
datos_sectores = []

# Se itera sobre todos los sectores identificados en orden
for sector in sorted(documentos_aleatorios["sector"].unique().tolist()):
    print(f"Evaluando el sector {sector}...")

    # Información requerida para el reporte
    total_documentos = documentos_aleatorios[documentos_aleatorios["sector"] == sector].shape[0]
    peso_total_sector = documentos_aleatorios[documentos_aleatorios["sector"] == sector]["totalPesoDocumento"].sum()
    volumen_total_sector = documentos_aleatorios[documentos_aleatorios["sector"] == sector]["totalVolumenDocumento"].sum()

    # Reporte de Los resultados de la agrupación en clústeres
    print("El total de documentos por despacharse en el sector es de:", total_documentos)
    print("El peso total por despacharse en el sector es de:", peso_total_sector)
    print("El volumen total por despacharse en el sector es de:", round(volumen_total_sector, 2), end="\n\n")

    # Se crea un diccionario que será utilizado para almacenar los datos del sector
    datos_sector = {
        "sector": sector,
        "total_documentos": total_documentos,
        "peso_total": peso_total_sector,
        "volumen_total": volumen_total_sector
    }
    # Se agrega el diccionario a la lista de los datos con los que se creará el nuevo dataframe
    datos_sectores.append(datos_sector)

# Se crea el nuevo dataframe con los datos de los sectores
sectores = pd.DataFrame(datos_sectores)
```

Ilustración 72

Resultados de la visualización del reporte de los sectores de entrega

```
Evaluando el sector 0...
El total de documentos por despacharse en el sector es de: 31
El peso total por despacharse en el sector es de: 43640.0
El volumen total por despacharse en el sector es de: 934.65

Evaluando el sector 1...
El total de documentos por despacharse en el sector es de: 8
El peso total por despacharse en el sector es de: 16963.9
El volumen total por despacharse en el sector es de: 417.4

Evaluando el sector 2...
El total de documentos por despacharse en el sector es de: 10
El peso total por despacharse en el sector es de: 17903.0
El volumen total por despacharse en el sector es de: 368.7

Evaluando el sector 3...
El total de documentos por despacharse en el sector es de: 11
El peso total por despacharse en el sector es de: 24373.0
El volumen total por despacharse en el sector es de: 550.05

Evaluando el sector 4...
El total de documentos por despacharse en el sector es de: 7
El peso total por despacharse en el sector es de: 11388.0
El volumen total por despacharse en el sector es de: 228.75
```

Ahora, con respecto a los sectores, se puede observar que los distintos clústeres identificados poseen cantidades, pesos y volúmenes diferentes, no obstante, teniendo en cuenta que LUMA procederá con la planificación de la logística de última milla de los sectores con menores capacidades con respecto al peso, entonces, el algoritmo comenzará con la automatización de la logística del sector 4, luego del 1, 2, 3, y, finalmente, del 0.

De esta manera, aplicando el algoritmo de asignación de vehículos a sectores y de documentos de cada sector a dichos vehículos, se obtuvieron los siguientes resultados:

Ilustración 73

Reporte de resultados generales

```
'''
Reporte de resultados generales
'''
# Información para el reporte
fecha_de_los_documentos = documentos_aleatorios["fechaEntrega"].unique()[0]
sectores = len(documentos_aleatorios["sector"].unique())
total_documentos_despachados = documentos_aleatorios[documentos_aleatorios["estadoDocumento"] == 4].shape[0]
total_vehiculos_utilizados = vehiculos[vehiculos["estadoVehiculo"] == "N"].shape[0]
total_peso_despachado = documentos_aleatorios[documentos_aleatorios["estadoDocumento"] == 4]["totalPesoDocumento"].sum()
total_volumen_despachado = documentos_aleatorios[documentos_aleatorios["estadoDocumento"] == 4]["totalVolumenDocumento"].sum()

# Reporte de información general
print(f"LUMA ha realizado la planificación de los documentos emitidos el día {fecha_de_los_documentos}")
print("Donde...")
print(f"\t-LUMA ha identificado {sectores} sectores de entrega.")
print(f"\t-LUMA ha planificado el despacho de {total_documentos_despachados} documentos.")
print(f"\t-LUMA ha asignado documentos a {total_vehiculos_utilizados} vehiculos.")
print(f"\t-LUMA ha despachado un total de peso de: {total_peso_despachado}")
print(f"\t-LUMA ha despachado un total de peso de: {total_volumen_despachado}")
```

```
LUMA ha realizado la planificación de los documentos emitidos el día 2024-03-22 00:00:00
Donde...
-LUMA ha identificado 5 sectores de entrega.
-LUMA ha planificado el despacho de 33 documentos.
-LUMA ha asignado documentos a 9 vehiculos.
-LUMA ha despachado un total de peso de: 68758.6
-LUMA ha despachado un total de peso de: 1497.9
```

Ilustración 74

Reporte de resultados específicos

```
'''
Reporte de resultados específicos
'''
for id_vehiculo in vehiculos["id_vehiculo"].unique():
    # Se identifica el total de peso y volumen que lleva el vehículo considerado
    total_peso = documentos_aleatorios[documentos_aleatorios["id_vehiculo"] == id_vehiculo]["totalPesoDocumento"].sum()
    total_volumen = documentos_aleatorios[documentos_aleatorios["id_vehiculo"] == id_vehiculo]["totalVolumenDocumento"].sum()

    # Se identifica la capacidad del vehículo en cuanto a peso y volumen
    capacidad_peso = vehiculos[vehiculos["id_vehiculo"] == id_vehiculo]["capacidadVehiculo"].iloc[0]
    capacidad_volumen = vehiculos[vehiculos["id_vehiculo"] == id_vehiculo]["capacidadVolumen"].iloc[0]

    # Se calculan los porcentajes de subutilización
    subutilizacion_peso = (capacidad_peso - total_peso) * 100 / capacidad_peso
    subutilizacion_volumen = (capacidad_volumen - total_volumen) * 100 / capacidad_volumen

    # Se reportan los resultados
    print(f"El vehículo {id_vehiculo}...")
    print(f"\t- Porcentaje de subutilización con respecto al peso: {round(subutilizacion_peso, 2)}%")
    print(f"\t- Porcentaje de subutilización con respecto al volumen: {round(subutilizacion_volumen, 2)}%", end="\n\n")
```

Ilustración 75

Resultados del reporte específico

El vehículo 113...
-Porcentaje de subutilización con respecto al peso: 8.75%
-Porcentaje de subutilización con respecto al volumen: 0.79%

El vehículo 115...
-Porcentaje de subutilización con respecto al peso: 34.0%
-Porcentaje de subutilización con respecto al volumen: 21.77%

El vehículo 117...
-Porcentaje de subutilización con respecto al peso: 0.26%
-Porcentaje de subutilización con respecto al volumen: 0.0%

El vehículo 111...
-Porcentaje de subutilización con respecto al peso: 2.62%
-Porcentaje de subutilización con respecto al volumen: 1.88%

El vehículo 114...
-Porcentaje de subutilización con respecto al peso: 63.81%
-Porcentaje de subutilización con respecto al volumen: 56.52%

El vehículo 112...
-Porcentaje de subutilización con respecto al peso: 23.38%
-Porcentaje de subutilización con respecto al volumen: 2.33%

El vehículo 116...
-Porcentaje de subutilización con respecto al peso: 23.38%
-Porcentaje de subutilización con respecto al volumen: 2.33%

El vehículo 118...
-Porcentaje de subutilización con respecto al peso: 1.67%
-Porcentaje de subutilización con respecto al volumen: 3.33%

El vehículo 119...
-Porcentaje de subutilización con respecto al peso: 45.0%
-Porcentaje de subutilización con respecto al volumen: 45.83%

Ilustración 76

Reporte de subutilizaciones

```
'''  
Reporte de subutilizaciones  
'''  
  
# Se resumen los porcentajes de subutilización  
resumen_datos_nuevos_con_max = {  
    "Categoría de uso": ["Mayor parte del uso", "Uso promedio", "Máximo reportado"],  
    "Subutilización Volumen (%)": [percentil_75_nuevo, 14.98, max_volumen],  
    "Subutilización Peso (%)": [percentil_75_peso, 22.54, max_peso]  
}  
  
# Se crea un DataFrame para el resumen  
df_resumen_con_max = pd.DataFrame(resumen_datos_nuevos_con_max)  
  
# Se visualiza el resumen de las subutilizaciones  
df_resumen_con_max
```

| | Categoría de uso | Subutilización Volumen (%) | Subutilización Peso (%) |
|---|---------------------|----------------------------|-------------------------|
| 0 | Mayor parte del uso | 21.77 | 34.00 |
| 1 | Uso promedio | 14.98 | 22.54 |
| 2 | Máximo reportado | 56.52 | 63.81 |

De tal modo, se puede observar que LUMA ha logrado la planificación de la logística de última milla de manera automática de 33 documentos haciendo uso de los 9 vehículos disponibles en la flota, lo que significa que planificó el despacho de 68758.6 Kg y 1497.9QQ de mercancías.

Además, el porcentaje más alto de subutilización reportado fue de 63.81% con respecto al peso y de 56.52% con relación al volumen, mientras que, considerando el promedio, hubo un 22.54% de subutilización relacionado al peso y tan solo un 14.97% con respecto al volumen; además, correspondiéndose a la mayor parte del uso, se identificó un porcentaje de subutilización del 21.77% con respecto al volumen y un 34% para el peso.

Esto, se puede comparar a los hallazgos referentes a la subutilización de vehículos identificada durante la exploración de datos mediante los gráficos a continuación:

Ilustración 77

Comparación de porcentajes de subutilización con respecto al peso

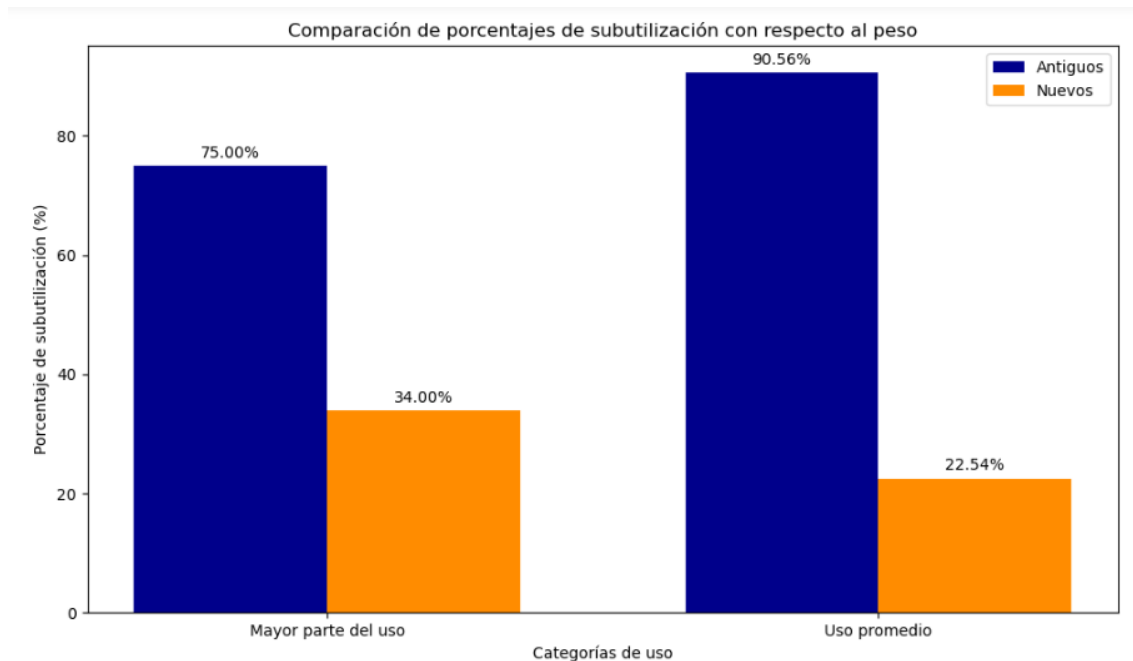
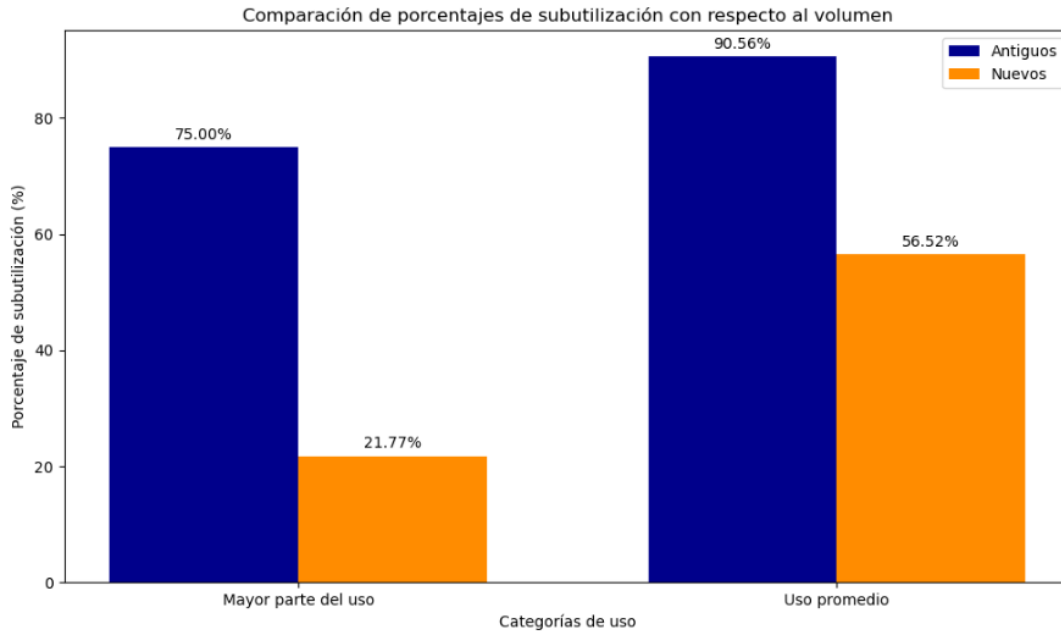


Ilustración 78

Comparación de porcentajes de subutilización con respecto al volumen



De este modo, se evidencia una clara reducción en los porcentajes de subutilización con respecto a la mayor parte del uso, con una disminución de hasta aproximadamente un 50% con relación al volumen y del 40% para el peso, mientras que, con respecto al promedio, existe una reducción del porcentaje de subutilización desde un 70% con relación al peso y de un 40% para el volumen.

En conclusión, se puede determinar que el desarrollo de una logística de última milla automatizada se ha logrado con éxito, pues, se ha alcanzado el objetivo de negocio exitosamente.

CAPÍTULO IX: CONCLUSIONES Y RECOMENDACIONES

9. Conclusiones y recomendaciones

9.1. Conclusiones

Para empezar, se puede concluir que la automatización de la asignación de productos a vehículos de despacho en la logística de última milla mediante técnicas de inteligencia artificial ha sido un éxito, pues, si bien el criterio de éxito del trabajo era el reducir la subutilización de los vehículos en un 40%, LUMA logró decrecer este porcentaje de subutilización desde un 40% hasta un 70%, por lo que, se puede determinar que se ha logrado mejorar la asignación de documentos a vehículos de despacho.

Por otro lado, se puede determinar que se logró analizar y comprender el proceso de asignación de productos a vehículos en la logística de última milla, dicho proceso fue sintetizado en un diagrama de flujo y permitió sentar las bases con las que se desarrollaría LUMA.

Además, también se logró elaborar un plan que permitió automatizar la logística de última milla mediante la aplicación de aprendizaje automático no supervisado, en específico, haciendo uso de la técnica de agrupación en clústeres.

En este sentido, de igual manera se concluyó el desarrollo de un modelo de agrupación en clústeres mediante KMeans que permitió identificar sectores de entrega para cualquier número de documentos que se emitan en un día.

Finalmente, también se logró desarrollar un algoritmo capaz de asignar de manera óptima vehículos de despacho a los sectores de entrega identificados por parte del modelo de agrupación en clústeres, el cual, consiste en 4 partes fundamentales.

9.2. Recomendaciones

Para futuras investigaciones sobre la automatización de la logística de última milla, se recomienda utilizar un enfoque similar al presentado en este trabajo, donde, se utiliza aprendizaje automático no supervisado, puesto que, si bien actualmente esta técnica es utilizada con varios fines, no se ha presenciado un uso similar en el que se hace alusión al estilo de planificación que realizaría un humano común y corriente.

En futuras implementaciones, se recomienda añadir condiciones adicionales, como la inclusión de ventanas horarias y restricciones vehiculares, como las que se aplican durante el Pico y Placa en Quito.

Finalmente, se recomienda hacer de LUMA un API que pueda recibir, a través de tipos de archivo JSON, los conjuntos de datos de Documentos y Vehículos, para que, posteriormente, LUMA los procese y devuelva los documentos con sus respectivos vehículos asignados y estados en el mismo formato.

BIBLIOGRAFÍA

- Bopage, G., Vidanagamachchi, K., & Julian, N. (2019). A strategic model to Improve the Last mile delivery performance in Ecommerce Parcel Delivery. *IEOM Society*, 14. Obtenido de <https://www.ieomsociety.org/ieom2019/papers/531.pdf>
- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinarts, T., Shearer, C., & Wirth, R. (1999). *CRISP-DM 1. Step-by-step data mining guide*. SPSS.
- DispatchTrack. (1 de noviembre de 2022). *Qué es la logística de última milla y por qué es clave para tu negocio*. Obtenido de DispatchTrack: <https://www.dispatchtrack.com/es/blog/que-es-la-logistica-de-ultima-milla#:~:text=En%20t%C3%A9rminos%20simples%2C%20la%20log%C3%ADstica,tambi%C3%A9n%20es%20el%20m%C3%A1s%20costoso>
- Eriklinderman. (25 de septiembre de 2023). *Overview for Microsoft Azure Maps*. Obtenido de Microsoft Learn: <https://learn.microsoft.com/en-us/azure/azure-maps/about-azure-maps>
- Gevaers, R., Van de Voorde, E., & Vanellander, T. (2014). Cost Modelling and Simulation of Last-mile Characteristics in an Innovative B2C Supply Chain Environment with Implications on Urban Areas and Cities. *Procedia - Social and Behavioral Sciences*, 14. doi:<https://doi.org/10.1016/j.sbspro.2014.01.1483>
- Giuffrida, N., Fajard Calderin, J., Masegosa, A., Werner, F., Steudter, M., & Pilla, F. (2022). *Optimization and Machine Learning Applied to Last-Mile Logistics: A Review*.

Sustainability, 14(9)(5329), 16. Obtenido de <https://www.mdpi.com/2071-1050/14/9/5329>

Gutierrez Rodríguez, A. E., Conant Pablos, S. E., Ortiz Bayliss, J. C., & Terashima Marín, H. (2019). Selecting meta-heuristics for solving vehicle routing problems with time windows via meta-learning. *Expert Systems With Applications*, 118, 12. doi:<https://doi.org/10.1016/j.eswa.2018.10.036>

Hanne, T. (2009). Reviewed Work(s): The Vehicle Routing Problem: Latest Advances and New Challenges by B. Golden, S. Raghavan and E. Wasil. *The Journal of the Operational Research Society*, 60(11), 3. Obtenido de <http://www.jstor.org/stable/40295725>

IBM. (s.f.). *¿Qué es el aprendizaje no supervisado?* Obtenido de IBM: <https://www.ibm.com/es-es/topics/unsupervised-learning>

IBM. (agosto de 2021). *CRISP-DM Help Overview*. Obtenido de IBM Documentation: <https://www.ibm.com/docs/en/spss-modeler/saas?topic=dm-crisp-help-overview>

INESDI. (2 de noviembre de 2022). *¿Qué es el aprendizaje no supervisado?* Obtenido de INESDI: <https://www.inesdi.com/blog/que-es-aprendizaje-no-supervisado/>

Jucha, P. (2021). Use of artificial intelligence in last mile delivery. *SHS Web of Conferences*, 92, 9. doi:<https://doi.org/10.1051/shsconf/20219204011>

Keytrends. (18 de abril de 2024). *Clustering o Agrupación de datos*. Obtenido de Keeytrends: <https://keytrends.ai/es/academy/glosario/inteligencia-artificial/clustering#:~:text=Se%20pueden%20distinguir%20dos%20tipos,subconjuntos%20cada%20vez%20m%C3%A1s%20peque%C3%B1os.>

- Lim, S., Xin Jin, & Singh Srai, J. (2015). Last-mile logistics models: A literature review and design guideline. *Institute for Manufacturing, Cambridge University*, 9. Obtenido de https://www.researchgate.net/publication/281783277_Last-mile_logistics_models_A_literature_review_and_design_guideline
- Llegó. (24 de abril de 2023). *Los 4 pasos para tener una logística de última milla de impacto*. Obtenido de Llegó: <https://blog.llego.cl/los-4-pasos-para-tener-una-log%C3%ADstica-de-%C3%BAltima-milla-de-impacto>
- López, E., & Flórez, N. (2023). Evolución de la logística de la última milla: Revisión de la literatura. *Ingeniería Industrial, XLIV*(2), 15. Obtenido de <http://scielo.sld.cu/pdf/rii/v44n2/1815-5936-rii-44-02-216.pdf>
- Mor, A., & Speranza, M. (2020). Vehicle routing problems over time: a survey. *4OR*, 18(2), 32. doi:<https://doi.org/10.1007/s10288-020-00433-2>
- Morash, E., Droge, C., & Vickery, S. (1996). Strategic logistics capabilities for competitive advantage and firm success. *Journal of Business Logistics*, 23. Obtenido de <https://eds.p.ebscohost.com/eds/detail/detail?vid=1&sid=0f6db224-d6b4-4de0-b700-f39f2df3dbe1%40redis&bdata=Jmxhbmc9ZXMmc2l0ZT1lZHMtbGl2ZQ%3d%3d#db=bth&AN=9708152650>
- Open Data Science. (12 de diciembre de 2021). *AI as the Ultimate Disrupter in Logistics: How to Manage Last-Mile Costs?* Obtenido de Medium: <https://odsc.medium.com/ai-as-the-ultimate-disrupter-in-logistics-how-to-manage-last-mile-costs-c4874e8f2ea0>
- Ousaid, H. (2020). Soluciones logísticas en la última milla. *[Tesis de Maestría]*. Universidad de Valladolid. Obtenido de <http://uvadoc.uva.es/handle/10324/41721>

- Patel, R. (12 de enero de 2024). *Machine Learning in Route Optimization: Techniques & Benefits*. Obtenido de Upper Route Planner: <https://www.upperinc.com/blog/machine-learning-route-optimization/>
- Peralta, F. (29 de junio de 2023). *Logística de Última Milla e Inteligencia Artificial: Reduciendo CO2*. Obtenido de Driv.In: <https://driv.in/blog/inteligencia-artificial-ultima-milla-reducir-co2#:~:text=Inteligencia%20artificial%20en%20la%20%C3%BAltima%20Milla,-Adem%C3%A1s%20de%20la&text=La%20inteligencia%20artificial%20desempe%C3%B1a%20un,de%20planificaci%C3%B3n%20manuales%20o>
- Rosendorff, A., Hodes, A., & Fabian, B. (2021). Artificial Intelligence for Last-mile logistics - Procedures and architecture. *Online Journal of Applied Knowledge Management*, 9(1), 16. Obtenido de <https://d-nb.info/1248645642/34>
- SimpliRoute. (s.f.). *Planificación de Rutas: Cómo Hacerlo de Forma Eficiente*. Obtenido de SimpliRoute: <https://simpliroute.com/es/blog/planificacion-de-rutas>
- Telefónica. (21 de septiembre de 2023). *¿Qué es el aprendizaje no supervisado?* Obtenido de Telefónica: <https://www.telefonica.com/es/sala-comunicacion/blog/que-es-aprendizaje-no-supervisado/>
- Universidad Europea. (11 de octubre de 2022). *Aprendizaje supervisado y no supervisado*. Obtenido de Universidad Europea: <https://universidadeuropea.com/blog/aprendizaje-supervisado-no-supervisado/>