

Trabajo Fin de Grado
Grado en Ingeniería en Tecnologías Industriales

**Aplicación del VRP en la red de logística de una
empresa del sector del automóvil**

Autor: Jorge Ruiz Montes de Oca
Tutor: Jose Miguel León Blanco

Dpto. Organización Industrial y
gestión de empresas I
Escuela Técnica Superior de
Ingeniería

Universidad de Sevilla

Sevilla, 2025



Trabajo Fin de Grado

Grado en Ingeniería en Tecnologías Industriales

Aplicación del VRP en la red de logística de una empresa del sector del automóvil

Autor:

Jorge Ruiz Montes de Oca

Tutor:

Jose Miguel León Blanco

Dpto. de Organización y gestión de

empresas I

Escuela Técnica Superior de

Ingeniería

Universidad de Sevilla

Sevilla, 2025

Trabajo Fin de Grado: Aplicación del VRP en la red de logística de una empresa del sector del automóvil

Autor: Jorge Ruiz Montes de Oca

Tutor: Jose Miguel León Blanco

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

*A mis padres, porque siempre me han dado todo y
más para que yo pueda realizar mis sueños.*

*A mis amistades, por apoyarnos tantas horas en la
biblioteca y hacer de Sevilla mi hogar.*

*Este trabajo no es el final, sino el comienzo de un
proyecto personal ambicioso, con el que
espero poder devolver a mis seres queridos
todo lo recibido estos años.*

Resumen

Este trabajo tiene como objetivo la aplicación del Problema de Enrutamiento de Vehículos (VRP) a la red logística de una empresa especializada en el suministro de recambios de automóviles. Se analizará la distribución de pedidos desde su sede hacia los diferentes talleres dentro de una ciudad y sus alrededores, optimizando las rutas de reparto mediante dos enfoques algorítmicos: Google OR-Tools para resolver el VRP y Recocido Simulado (Simulated Annealing, SA) para refinar la solución inicial.

Se estudiará el caso en el que un máximo de tres vehículos realice el reparto en ciclos de 55 minutos, asegurando que se atiendan los pedidos en el menor tiempo posible y con la menor cantidad de vehículos. Para ello, se utilizará una matriz de distancias preexistente entre los talleres, de la cual se seleccionarán dinámicamente los clientes a atender en cada turno.

Nos encontramos ante un problema de optimización complejo pero fundamental en la logística moderna, donde la reducción de kilómetros recorridos, tiempo de entrega y costes operativos representa un gran desafío empresarial. Con este estudio se busca proporcionar una solución eficiente que pueda ser aplicada en la operativa real de la empresa, mejorando la planificación de rutas y la gestión de flotas.

Abstract

This study aims to apply the Vehicle Routing Problem (VRP) to the logistics network of a company specializing in the supply of automobile spare parts. The distribution of orders from its central location to various workshops within a city and its surroundings, optimizing delivery routes using two algorithmic approaches: Google OR-Tools to solve the VRP and Simulated Annealing (SA) to refine the initial solution.

The study will consider the case where a maximum of three vehicles performs deliveries in 55-minute cycles, ensuring that orders are fulfilled in the shortest possible time while minimizing the number of vehicles used. To achieve this, a pre-existing distance matrix between the workshops will be utilized, dynamically selecting the customers to be served in each delivery cycle.

This research addresses a complex yet essential optimization problem in modern logistics, where reducing travel distance, delivery time, and operational costs represents a significant business challenge. The goal of this study is to provide an efficient solution that can be implemented in the company's real-world operations, enhancing route planning and fleet management.

Índice

Resumen	X
Abstract	XII
1 Introducción	1
1.1. Contexto y motivación	1
1.2. Objetivos del trabajo	2
1.3. Estructura del trabajo	3
2 La empresa y su logística	4
2.1. Estructura organizativa y modelo de negocio.....	4
2.2. Logística de distribución en la empresa.....	5
2.2.1. Infraestructura logística	5
2.2.2. Procesos logísticos.....	6
2.3. Desafíos y problemáticas en la distribución	6
2.3.1. Ineficiencias en la planificación de rutas.....	6
2.3.2. Restricciones operativas	7
2.3.3. Oportunidades de mejora.....	7
3 Fundamentos teóricos del problema de enrutamiento de vehículos (VRP).....	8
3.1. Introducción	8
3.2. Definición formal del Problema de Enrutamiento de Vehículos (VRP).....	8
3.3. Variantes del VRP	10

3.3.1.	VRP con capacidad limitada (Capacitated VRP - CVRP)	11
3.3.2.	VRP con ventanas de tiempo (VRPTW - Vehicle Routing Problem with Time Windows).....	11
3.3.3.	VRP con múltiples almacenes (Multi-Depot VRP - MDVRP).....	12
3.3.4.	VRP con costos heterogéneos	12
3.3.5.	VRP dinámico (Dynamic VRP - DVRP)	12
3.3.6.	VRP con pedidos divididos (Split Delivery VRP - SDVRP).....	13
3.3.7.	VRP con restricciones de combustible o baterías (Green VRP - GVRP) ...	13
3.3.8.	VRP seleccionado para este estudio	13
3.4.	<i>Métodos de resolución del VRP</i>	15
3.4.1.	Métodos exactos	16
3.4.1.1.	Programación Entera Mixta (MIP).....	16
3.4.2.	Métodos heurísticos.....	16
3.4.1.2.	Algoritmo de Ahorro de Clarke & Wright	16
3.4.3.	Métodos metaheurísticos	17
3.4.1.3.	Recocido Simulado (Simulated Annealing - SA).....	17
3.4.4.	Uso de Google OR-Tools para la resolución del VRP	18
3.5.	<i>Conclusión</i>	18
4 Modelos Algorítmicos Aplicados al VRP		19
4.1.	<i>Introducción</i>	19
4.2.	<i>Google OR-Tools para VRP</i>	19

4.2.1.	Fundamentos de OR-Tools.....	19
4.2.2.	Enfoque utilizado en OR-Tools para el VRP	19
4.2.3.	Limitaciones de OR-Tools	20
4.3.	<i>Recocido Simulado (Simulated Annealing - SA)</i>	20
4.3.1.	Principios del Recocido Simulado.....	20
4.3.2.	Aplicación del Recocido Simulado en el VRP.....	21
4.3.3.	Ventajas y desventajas del Recocido Simulado	21
4.4.	<i>Comparación entre Google OR-Tools y Recocido Simulado</i>	22
4.5.	<i>Conclusión</i>	22
5	Propuesta de Implementación Detallada de los Modelos Teóricos.....	24
5.1.	<i>Obtención de Coordenadas Geográficas</i>	24
5.1.1.	Fundamentos Técnicos	24
5.1.2.	Implementación con la API de Google Maps.....	24
5.2.	<i>Generación de la Matriz de Distancias y Tiempos</i>	25
5.2.1.	Importancia de la Matriz	25
5.2.2.	Creación de la Matriz con Google Maps Distance Matrix API.....	25
5.3.	<i>Preparación del Problema VRP</i>	26
5.3.1.	Estructura del Problema	26
5.3.2.	Configuración del Problema en Código	26
5.4.	<i>Implementación de OR-Tools</i>	27
5.4.1.	Justificación del Algoritmo	27

5.4.2.	Configuración del Modelo.....	27
5.4.3.	Implementación Paso a Paso	27
5.4.4.	Resultados y Ventajas.....	28
5.5.	<i>Implementación del Recocido Simulado (Simulated Annealing)</i>	28
5.5.1.	Justificación del Algoritmo	28
5.5.2.	Parámetros del Recocido Simulado.....	28
5.5.3.	Implementación Paso a Paso	29
5.6.	<i>Integración Final</i>	29
5.7.	<i>Consideraciones de Seguridad y Escalabilidad</i>	30
6	Propuesta de Implementación del Software.....	31
6.1.	<i>Arquitectura General del Sistema</i>	31
6.2.	<i>Requisitos de Software y Herramientas</i>	32
6.3.	<i>Flujo Detallado de Procesamiento de Datos</i>	32
6.4.	<i>Estructura de Archivos del Proyecto</i>	33
6.5.	<i>Interfaz de Usuario</i>	33
6.6.	<i>Gestión de Errores y Excepciones</i>	35
6.7.	<i>Optimización del Rendimiento</i>	35
6.8.	<i>Despliegue en la Nube</i>	35
7	Impacto Empresarial de la Propuesta Tecnológica	37
7.1.	<i>Mejora de la eficiencia operativa y reducción de costes</i>	37
7.2.	<i>Sostenibilidad y responsabilidad medioambiental</i>	38

7.3.	<i>Mejora de la experiencia del cliente</i>	38
7.4.	<i>Escalabilidad, flexibilidad y adaptabilidad</i>	39
7.5.	<i>Impacto organizacional y cultural</i>	39
8	Conclusiones	41
	<i>Bibliografía</i>	44

1 Introducción

1.1. Contexto y motivación

En el panorama actual del comercio electrónico, la logística de última milla se ha convertido en un componente crítico de la cadena de suministro, representando hasta el 53% del costo total de envío [1]. Esta fase final, que abarca desde el centro de distribución hasta el cliente final, es indispensable para garantizar la satisfacción del cliente y la eficiencia operativa.

La empresa de estudio ha desempeñado un papel destacado en el sector de la posventa de automoción desde su fundación. Inicialmente enfocada en la comercialización de motocultores, la empresa ha experimentado un crecimiento constante a lo largo de más de cincuenta años, consolidándose como líder en la distribución de recambios y servicios para talleres de reparación de vehículos.

Con una red de delegaciones distribuidas en Andalucía, la empresa ofrece cobertura a 744 municipios, lo que representa aproximadamente el 25% del territorio nacional. Este alcance geográfico ha sido posible gracias a una estrategia de expansión sostenida y a una inversión continua en infraestructura logística y tecnológica.

La oficina de la empresa situada en un punto de Andalucía es una pieza clave en su estructura operativa, gestionando la distribución de productos. Esta delegación se encarga de coordinar las rutas de reparto para garantizar entregas eficientes y puntuales a sus clientes, que incluyen talleres mecánicos y otros profesionales del sector automotriz.

En 2019, la empresa alcanzó un volumen de ventas de 64 millones de euros, demostrando su solidez financiera y capacidad de adaptación en un mercado competitivo. Este éxito se atribuye en parte a su equipo humano, compuesto por casi 430 empleados comprometidos con la excelencia en el servicio y la satisfacción del cliente.

La planificación de rutas en la oficina de la provincia comentada se ha realizado tradicionalmente de manera manual, basándose en la experiencia de los conductores y en métodos convencionales. Sin embargo, este enfoque puede dar lugar a ineficiencias en

términos de distancia recorrida, consumo de combustible y cumplimiento de los horarios de entrega. Por lo tanto, existe una necesidad imperante de implementar soluciones de optimización que permitan mejorar la eficiencia operativa y reducir los costos asociados al transporte.

El Problema de Enrutamiento de Vehículos (VRP, por sus siglas en inglés) se presenta como una herramienta matemática adecuada para abordar este desafío. Este problema busca determinar la ruta más corta que permite a unos vendedores (o, en este caso, a unos vehículos de reparto) visitar una serie de ciudades (clientes) y regresar al punto de partida. Aplicar algoritmos de resolución del VRP en el contexto de dicha empresa podría traducirse en rutas de reparto más eficientes, optimizando recursos y mejorando la satisfacción del cliente.

1.2. Objetivos del trabajo

El objetivo principal de este TFG es optimizar las rutas de reparto de la oficina de la empresa estudiada en una de sus provincias mediante la aplicación de algoritmos de resolución del VRP. Para alcanzar este objetivo, se plantean los siguientes objetivos específicos:

Modelar matemáticamente el problema de las rutas de reparto, considerando las particularidades de la operación logística de la empresa.

Implementar y comparar dos algoritmos de resolución del VRP:

Google OR-Tools: una biblioteca de optimización desarrollada por Google que permite resolver el Problema de Enrutamiento de Vehículos (VRP) utilizando Programación Entera Mixta (MIP) y enfoques heurísticos avanzados. Su método base emplea el algoritmo de Búsqueda Local con Metaheurísticas para encontrar soluciones eficientes en tiempos razonables, combinando estrategias como Path Cheapest Arc y Guided Local Search (GLS) para optimizar las rutas de entrega.

Recocido Simulado (SA): una metaheurística que busca evitar óptimos locales mediante una exploración probabilística del espacio de soluciones.

Evaluar el desempeño de los algoritmos en términos de indicadores clave de

rendimiento (KPIs), como la distancia total recorrida y el tiempo de entrega.

Proponer recomendaciones para la implementación práctica de las soluciones en la operación diaria de la organización.

1.3. Estructura del trabajo

Este documento se organiza en los siguientes capítulos:

Capítulo 1: se plantea el contexto en el que el proyecto se plantea, siendo este la logística dentro de la red de la empresa

Capítulo 2: Se describe detalladamente a la organización, su historia, estructura organizativa y la logística actual de la oficina que se va a analizar.

Capítulo 3: Se introduce el Problema de Enrutamiento de Vehículos (VRP), su formulación matemática y su relevancia en problemas de optimización de rutas.

Capítulo 4: Se presentan las metodologías de resolución del VRP, enfocándose en los algoritmos de Google OR-Tools y Recocido Simulado.

Capítulo 5: Se detalla la aplicación práctica de los algoritmos en el contexto de las rutas de reparto de la empresa, incluyendo la obtención y procesamiento de datos reales.

Capítulo 6: Se analizan los resultados obtenidos, comparando el desempeño de los algoritmos y evaluando su aplicabilidad práctica. Se plantea una propuesta de implementación del software que permitiría trasladar los modelos teóricos y los algoritmos desarrollados a una herramienta práctica. Como parte de esta propuesta, se sugiere el uso de Streamlit, una librería de Python que facilita la creación de interfaces web de manera sencilla e interactiva. Esta interfaz permitiría, de forma potencial, la carga de datos, la ejecución de la optimización de rutas y la visualización de resultados sobre mapas, acercando la solución al entorno real de uso por parte de operarios o gestores logísticos.

Capítulo 7: Se concluye el trabajo con una síntesis de los hallazgos y se proponen líneas futuras de investigación y mejora.

2 La empresa y su logística

La organización estudiada es una empresa andaluza con más de 50 años de trayectoria en la distribución de recambios de automóviles. Inició su actividad con la comercialización de motocultores y maquinaria agrícola. Con el tiempo, la empresa diversifica su oferta, especializándose en la venta de repuestos para automóviles y consolidándose como una de las principales empresas de distribución de recambios en el sur de España.

Desde su fundación, la empresa ha seguido una estrategia de crecimiento sostenido, caracterizada por:

Expansión territorial, pasando de un negocio local a contar con 29 delegaciones en toda España.

Ampliación de catálogo, incorporando nuevas líneas de productos como baterías, lubricantes, neumáticos y herramientas para talleres.

Digitalización y automatización, integrando tecnologías avanzadas para la gestión de inventarios y optimización logística.

A lo largo de los años, la empresa ha mantenido una visión enfocada en la innovación y el servicio al cliente, ofreciendo asesoramiento técnico especializado y una red de distribución eficiente. En 2019, la empresa alcanzó una facturación de 64 millones de euros, consolidándose como un actor clave en el mercado de la postventa automotriz.

2.1. Estructura organizativa y modelo de negocio

La organización se estructura como una empresa de distribución con una amplia red de delegaciones. Su modelo de negocio se basa en el abastecimiento rápido y eficiente de talleres mecánicos, concesionarios y empresas del sector automotriz, asegurando que los repuestos lleguen en el menor tiempo posible.

- Principales características de su modelo de negocio:

Clientes principales: Talleres independientes, concesionarios oficiales, grandes flotas

y empresas de renting.

Catálogo de productos: Más de 100,000 referencias en stock, que incluyen repuestos mecánicos, eléctricos y de carrocería.

Red de distribución: 29 delegaciones.

Sistemas de gestión: Plataforma de pedidos online para agilizar la compra y distribución.

La delegación de la provincia seleccionada, foco de este estudio, es una de las más estratégicas, ya que atiende a clientes en toda la provincia, asegurando que los talleres tengan disponibilidad inmediata de los repuestos que necesitan.

2.2. Logística de distribución en la empresa

El éxito de la empresa se basa en su eficiencia logística, asegurando que los pedidos sean entregados en plazos reducidos para minimizar la inactividad de los talleres.

2.2.1. Infraestructura logística

El sistema de distribución de la empresa se apoya en centros de almacenamiento estratégicos y una flota de reparto que garantiza la entrega en tiempos optimizados.

Elementos clave de su infraestructura:

- **Almacenes regionales:** Cada delegación tiene un stock base que se repone desde el almacén central.
- **Flota de reparto:** Furgonetas de distintos tamaños para adaptarse a la demanda de los talleres.
- **Puntos de recogida:** En algunas delegaciones, los clientes pueden recoger sus pedidos directamente.

2.2.2. Procesos logísticos

El proceso de distribución sigue una estructura organizada para asegurar entregas en el menor tiempo posible:

- Recepción del pedido: Los talleres solicitan repuestos a través de la plataforma online o vía telefónica.
- Preparación en almacén: Los productos son recogidos en la delegación o enviados desde la ciudad principal si no están en stock local.
- Asignación de ruta de reparto: Se agrupan los pedidos según la zona geográfica para optimizar el recorrido.
- Entrega al cliente: Se priorizan los pedidos con ventanas de tiempo restringidas, asegurando que lleguen en el horario pactado.

Actualmente, la planificación de rutas se realiza en gran medida de forma manual, lo que puede generar ineficiencias en términos de distancia recorrida y tiempos de entrega.

2.3. Desafíos y problemáticas en la distribución

Pese a la eficiencia del modelo de la empresa, existen desafíos logísticos que afectan la distribución en la última milla, especialmente en una de sus provincias:

2.3.1. Ineficiencias en la planificación de rutas

Uno de los principales problemas es la **gestión manual de rutas**, que puede generar:

- Mayor número de kilómetros recorridos.
- Desviaciones innecesarias en los trayectos.
- Desigualdad en la carga de trabajo entre conductores.

Aplicar **algoritmos de optimización** puede reducir significativamente estos problemas.

2.3.2. Restricciones operativas

Otro factor clave son las restricciones logísticas que influyen en la planificación de rutas:

- **Capacidad de los vehículos:** La cantidad de repuestos que puede transportar cada furgoneta limita la asignación de pedidos a una misma ruta.
- **Ventanas de tiempo:** Algunos clientes requieren la entrega en horarios específicos, lo que condiciona el orden de visita de los talleres.
- **Accesibilidad:** Existen zonas urbanas con restricciones de tráfico o estacionamiento que pueden retrasar las entregas.

2.3.3. Oportunidades de mejora

Dada la naturaleza de estos desafíos, la optimización de rutas mediante algoritmos de Vecino Más Próximo y Recocido Simulado puede representar una solución viable para:

- Reducir la distancia recorrida y el consumo de combustible.
- Mejorar los tiempos de entrega.
- Distribuir mejor la carga de trabajo entre los vehículos.

3 Fundamentos teóricos del problema de enrutamiento de vehículos (VRP)

3.1. Introducción

El Problema de Enrutamiento de Vehículos (VRP, Vehicle Routing Problem) es una extensión del Problema del Viajante de Comercio (TSP, Travelling Salesman Problem), en el que no solo un único vehículo debe visitar una serie de puntos, sino que se dispone de varios vehículos que deben cubrir las entregas desde un depósito central hacia distintos clientes. Este problema es de gran relevancia en la optimización de rutas para la distribución de mercancías y servicios logísticos, siendo ampliamente estudiado en la investigación operativa y la inteligencia artificial.

El VRP fue formulado por primera vez en 1959 por Dantzig y Ramser, quienes lo aplicaron al reparto de combustible en camiones cisterna [2]. A lo largo de los años, se han desarrollado diversas variantes del VRP que incorporan restricciones de capacidad, tiempos de entrega y costos operacionales. Este problema es NP-difícil, lo que implica que no se puede resolver de manera exacta en tiempos razonables para instancias grandes, requiriendo el uso de algoritmos heurísticos y metaheurísticos [3].

3.2. Definición formal del Problema de Enrutamiento de Vehículos (VRP)

El VRP se define sobre un grafo dirigido $G = (V, A)$, donde:

- $V = \{v_0, v_1, \dots, v_n\}$ es el conjunto de nodos que representa los clientes y el depósito.
- A es el conjunto de aristas que conectan los nodos.
- d_{ij} representa la distancia o el costo de viajar entre los nodos i y j .
- Se dispone de K vehículos con una capacidad máxima Q .

El objetivo es encontrar un conjunto de rutas R_k para cada vehículo $k \in K$ tal que:

1. Cada cliente sea visitado exactamente una vez por un solo vehículo.

2. Cada ruta comience y termine en el depósito.
3. Se minimice la función objetivo, que puede ser la distancia total recorrida o el costo total del transporte.

x_{ij}^k Variable binaria:

- $x_{ij}^k=1$ si el vehículo k viaja del nodo i al nodo j
- $x_{ij}^k=0$ en caso contrario

d_{ij} Representa la distancia (o el coste) de viajar desde el nodo i hasta el nodo j .

Conjuntos:

- K : Conjuntos de vehículos disponibles.
- V : Conjunto de todos los nodos.
- $C \subset V$: Conjunto de clientes.
- El nodo 0 representa generalmente el almacén o depósito

A continuación, se presenta la formulación matemática del VRP:

(1) Función objetivo:

$$\min \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} d_{ij} x_{ij}^k \quad (1)$$

Significado: Minimiza la distancia total recorrida por todos los vehículos. Es decir, se busca la combinación de rutas que cubra todos los clientes usando la menor cantidad de kilómetros posibles.

Sujeto a:

(2) Restricción de cobertura de clientes:

$$\sum_{k \in K} \sum_{j \in V} x_{ij}^k = 1, \quad \forall i \in C \quad (2)$$

Significado: Cada cliente i debe ser visitado exactamente una vez por un solo vehículo, desde cualquier otro nodo j .

Donde x_{ij}^k es una variable binaria que indica si el vehículo k viaja desde el nodo i hasta el nodo j .

(3) Restricción de flujo:

$$\sum_{j \in V} x_{ij}^k = \sum_{j \in V} x_{ji}^k, \quad \forall k \in K, \forall i \in V \quad (3)$$

Significado: Por cada vehículo k , el número de veces que entra a un nodo debe ser igual al número de veces que sale de ese nodo. Mantiene el equilibrio de flujo en la ruta.

(4) Salida de almacén:

$$\sum_{j \in V} x_{0j}^k = 1, \quad \forall k \in K \quad (4)$$

Significado: Cada vehículo debe salir del depósito (nodo 0) exactamente una vez al inicio de su ruta.

(5) Retorno al almacén:

$$\sum_{i \in V} x_{i0}^k = 1, \quad \forall k \in K \quad (5)$$

Significado: Cada vehículo debe regresar al depósito (nodo 0) exactamente una vez al final de su ruta.

3.3. Variantes del VRP

El Problema de Enrutamiento de Vehículos (VRP) es un problema altamente flexible, que puede adaptarse a distintos escenarios dependiendo de las restricciones y condiciones del sistema de distribución. Con el tiempo, han surgido múltiples variantes del

VRP, cada una diseñada para abordar desafíos específicos en la optimización del transporte y la logística.

A continuación, se presentan las variantes más estudiadas en la literatura y sus principales características:

3.3.1. VRP con capacidad limitada (Capacitated VRP - CVRP)

El Capacitated VRP (CVRP) es una de las versiones más comunes del problema, en la cual cada vehículo tiene una capacidad máxima de carga y no puede excederla al asignar clientes en su ruta. Este tipo de problema es fundamental en la logística de distribución, especialmente en industrias como la alimentación, el transporte de bienes de consumo y la mensajería, donde los camiones tienen una capacidad fija.

Ejemplo de aplicación:

Una empresa de distribución de productos farmacéuticos debe planificar la entrega de medicamentos a distintas farmacias en una ciudad. Cada furgoneta tiene un límite de peso que no puede sobrepasar, por lo que el algoritmo de VRP debe optimizar no solo la distancia recorrida, sino también la carga transportada.

3.3.2. VRP con ventanas de tiempo (VRPTW - Vehicle Routing Problem with Time Windows)

En esta variante, cada cliente debe ser visitado dentro de un intervalo de tiempo predefinido. Es especialmente útil en entornos donde hay restricciones de disponibilidad de los clientes, como en la distribución de paquetes, el transporte de pasajeros y los servicios de mantenimiento.

Ejemplo de aplicación:

Una empresa de reparto de comida tiene franjas horarias en las que los clientes han solicitado recibir sus pedidos. Si un cliente ha programado su entrega entre las 12:00 y las 12:30, el algoritmo de optimización debe asegurarse de que el pedido llegue dentro de ese rango de tiempo.

3.3.3. VRP con múltiples almacenes (Multi-Depot VRP - MDVRP)

El Multi-Depot VRP (MDVRP) es una extensión del VRP en la que los vehículos pueden partir desde más de un centro de distribución. Este problema es particularmente relevante en empresas que tienen varias sedes logísticas y necesitan coordinar la distribución de mercancías desde distintas ubicaciones.

Ejemplo de aplicación:

Una cadena de supermercados con almacenes en diferentes ciudades debe organizar las rutas de reparto para minimizar costos y tiempos de entrega. En este caso, el algoritmo debe determinar desde qué depósito se debe asignar cada pedido y qué vehículo lo transportará.

3.3.4. VRP con costos heterogéneos

En esta variante, los costos de operación de cada vehículo son distintos. Esto puede deberse a diferencias en el tipo de combustible, el consumo energético, la velocidad promedio o el desgaste mecánico. Este problema es muy común en flotas que combinan vehículos eléctricos y de combustión, o en empresas de transporte con diferentes tipos de camiones.

Ejemplo de aplicación:

Una compañía de transporte utiliza una flota mixta de camiones diésel y eléctricos. Los camiones eléctricos tienen menor costo por kilómetro, pero necesitan hacer paradas de recarga, lo que afecta la planificación de las rutas.

3.3.5. VRP dinámico (Dynamic VRP - DVRP)

El VRP dinámico es aquel en el que las condiciones del problema cambian en tiempo real, como la incorporación de nuevos pedidos, modificaciones en las rutas debido al tráfico o cancelaciones de entregas. Esta variante es común en aplicaciones modernas que requieren respuestas en tiempo real, como en empresas de transporte de pasajeros y servicios de taxi.

Ejemplo de aplicación:

En una aplicación de transporte privado como Uber, las solicitudes de viaje van surgiendo continuamente, lo que obliga al algoritmo a reconfigurar las rutas y asignar nuevos viajes a los conductores disponibles en cada momento.

3.3.6. VRP con pedidos divididos (Split Delivery VRP - SDVRP)

En esta variante, un cliente puede ser atendido por más de un vehículo si su demanda excede la capacidad de un solo camión. Esto es útil en industrias donde los pedidos son grandes y es más eficiente dividir la entrega en varios viajes.

Ejemplo de aplicación:

Una fábrica de muebles necesita entregar 100 sillas a una tienda, pero sus camiones solo pueden transportar 50 sillas cada uno. El algoritmo debe dividir la entrega en dos viajes sin aumentar demasiado la distancia recorrida.

3.3.7. VRP con restricciones de combustible o baterías (Green VRP - GVRP)

El Green VRP (GVRP) considera el consumo de combustible o la autonomía de las baterías de los vehículos eléctricos. En este problema, es necesario planificar rutas optimizadas teniendo en cuenta las estaciones de recarga o los costos de consumo de combustible.

Ejemplo de aplicación:

Una empresa de logística que usa camiones eléctricos debe optimizar sus rutas asegurándose de que cada vehículo tenga acceso a una estación de carga antes de que se agote su batería.

3.3.8. VRP seleccionado para este estudio

Para este trabajo, se considerará un VRP con restricciones de tiempo (VRPTW), donde los vehículos deben completar su ruta en un máximo de 55 minutos, incluyendo

tiempos de carga y descarga en cada cliente. Esta variante se ajusta a las necesidades de la empresa, ya que su red de distribución opera bajo ciclos de reparto predefinidos, asegurando que las entregas sean rápidas y eficientes.

t_{ij} Tiempo necesario para que un vehículo viaje desde el nodo i al nodo j .

y_k Variable binaria que indica si el vehículo k ha sido utilizado:

- $y_k=1$: Si el vehículo está en uso.
- $y_k=0$: Si no se utiliza.

(6) Función objetivo

$$\min \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} d_{ij} x_{ij}^k \quad (6)$$

Igual que antes, se busca minimizar la distancia total recorrida por todos los vehículos.

(7) Cada cliente debe ser visitado una vez

$$\sum_{k \in K} \sum_{j \in J} x_{ij}^k = 1, \quad \forall i \in C \quad (7)$$

Cada cliente debe ser visitado exactamente una vez por un solo vehículo.

(8) Conservación de flujo

$$\sum_{j \in J} x_{ij}^k = \sum_{j \in V} x_{ji}^k, \quad \forall k \in K, \forall i \in V \quad (8)$$

Si un vehículo llega a un nodo, también debe salir de él. Mantiene rutas cerradas y consistentes

(9) Salida del depósito

$$\sum_{j \in V} x_{0j}^k = 1, \quad \forall k \in K \quad (9)$$

Cada vehículo sale una vez del almacén (nodo 0).

(10) Retorno al depósito

$$\sum_{i \in V} x_{i0}^k = 1, \quad \forall k \in K \quad (10)$$

Cada vehículo debe regresar al almacén exactamente una vez.

(11) Restricción de tiempo

$$\sum_{i \in V} \sum_{j \in V} t_{ij} x_{ij}^k + \sum_{i \in C} 3 \leq 55, \quad \forall k \in K \quad (11)$$

Cada vehículo tiene como máximo 55 minutos para completar su ruta, incluyendo el tiempo de viaje y 3 minutos por cada cliente (carga/descarga).

(12) Límite de vehículos utilizados

$$\sum_{k \in K} y_k \leq 3 \quad (12)$$

Solo se pueden utilizar como máximo 3 vehículos

3.4. Métodos de resolución del VRP

Dado que el Problema de Enrutamiento de Vehículos (VRP) es NP-difícil, su resolución óptima para instancias grandes es computacionalmente costosa. Por ello, se han desarrollado diversos métodos para encontrar soluciones en tipos razonables. Estos métodos se pueden clasificar en tres grandes grupos [6].

1. Métodos exactos
2. Métodos heurísticos

3. Métodos metaheurísticos

A continuación, se describen en detalle cada uno de estos enfoques, sus ventajas y limitaciones.

3.4.1. Métodos exactos

Los métodos exactos garantizan encontrar la solución óptima del VRP resolviendo todas las combinaciones posibles de rutas. Sin embargo, su tiempo de ejecución crece exponencialmente a medida que aumenta el número de clientes y vehículos [4].

3.4.1.1. Programación Entera Mixta (MIP)

Descripción:

La Programación Entera Mixta (MIP) consiste en modelar el VRP como un conjunto de variables, restricciones y una función objetivo para minimizar el costo total de las rutas [5].

Ejemplo de aplicación:

Empresas de logística que gestionan un número reducido de clientes y requieren la solución óptima garantizada, como el reparto de medicamentos urgentes a hospitales.

Limitaciones:

No es viable para instancias grandes, ya que el número de combinaciones crece de manera factorial.

3.4.2. Métodos heurísticos

Los algoritmos heurísticos son métodos aproximados que encuentran soluciones factibles en tiempos reducidos, aunque no garantizan que sean óptimas [6].

3.4.1.2. Algoritmo de Ahorro de Clarke & Wright

Descripción:

El algoritmo de Clarke & Wright calcula el ahorro en distancia cuando dos rutas

individuales se combinan en una sola [7].

Ejemplo de aplicación:

Empresas de distribución de alimentos que optimizan el uso de camiones para abastecer supermercados reduciendo kilómetros recorridos.

Ventajas:

- Rápido y fácil de implementar.
- Proporciona soluciones eficientes en problemas medianos.

Limitaciones:

No siempre encuentra la mejor solución posible.

3.4.3. Métodos metaheurísticos

Los metaheurísticos son algoritmos avanzados que mejoran soluciones heurísticas a través de estrategias de búsqueda más inteligentes [8].

3.4.1.3. Recocido Simulado (Simulated Annealing - SA)

Descripción:

Inspirado en el proceso de enfriamiento de metales, el Recocido Simulado (SA) explora soluciones cercanas e introduce perturbaciones aleatorias para evitar quedarse atrapado en un mínimo local.

Ejemplo de aplicación:

Empresas de transporte de carga que buscan reducir el número de camiones en ruta sin aumentar en exceso la distancia recorrida.

Ventajas:

- Puede mejorar significativamente las soluciones iniciales.
- Se adapta bien a problemas grandes.

Limitaciones:

- Requiere ajuste fino de parámetros (temperatura, iteraciones, enfriamiento).

3.4.4. Uso de Google OR-Tools para la resolución del VRP

En este trabajo, se utilizará Google OR-Tools, una biblioteca de optimización desarrollada por Google que combina diversas estrategias para resolver el VRP de forma eficiente [9].

Métodos empleados por OR-Tools:

- Path Cheapest Arc: Expande la ruta con el cliente más barato en cada paso.
- Local Search Metaheuristics: Aplica metaheurísticas como Guided Local Search (GLS) para mejorar soluciones.

Ventajas de OR-Tools:

- Alta velocidad de resolución.
- Fácil implementación con restricciones de tiempo y capacidad.
- Soporta integración con Python.

3.5. Conclusión

El Problema de Enrutamiento de Vehículos es fundamental en la optimización logística y la distribución de mercancías. En este trabajo, se abordará su resolución mediante Google OR-Tools y Recocido Simulado, buscando minimizar la distancia total recorrida y mejorar la eficiencia en la entrega de pedidos

4 Modelos Algorítmicos Aplicados al VRP

4.1. Introducción

En este capítulo se explicarán en detalle los algoritmos utilizados para resolver el Problema de Enrutamiento de Vehículos (VRP) en este estudio. Como se mencionó anteriormente, se empleará Google OR-Tools para obtener una solución inicial del problema y luego se aplicará Recocido Simulado (SA) para optimizar la calidad de la solución.

Ambos algoritmos ofrecen enfoques distintos para resolver el VRP: mientras que Google OR-Tools se basa en heurísticas avanzadas y metaheurísticas para encontrar soluciones de manera eficiente, el Recocido Simulado (SA) permite mejorar la calidad de estas soluciones al explorar mejor el espacio de búsqueda.

El objetivo de este capítulo es proporcionar una comprensión completa de los fundamentos teóricos de ambos métodos, sus principios de funcionamiento y sus ventajas y limitaciones en el contexto del VRP.

4.2. Google OR-Tools para VRP

4.2.1. Fundamentos de OR-Tools

Google OR-Tools es una biblioteca de optimización desarrollada por Google para resolver problemas de optimización combinatoria, como el Problema del Viajante de Comercio (TSP) y el Problema de Enrutamiento de Vehículos (VRP). OR-Tools permite modelar y resolver problemas logísticos aplicando diversas estrategias como búsqueda local, heurísticas y metaheurísticas avanzadas [6].

4.2.2. Enfoque utilizado en OR-Tools para el VRP

El proceso de resolución en OR-Tools sigue varios pasos clave:

- Definición del problema: Se establece una matriz de distancias entre nodos y se define la cantidad de vehículos disponibles.
- Selección de una estrategia de optimización: OR-Tools emplea heurísticas como Path Cheapest Arc para generar una primera solución factible, basada en la selección de la ruta con menor costo en cada iteración [9].
- Mejora de la solución con metaheurísticas: Una vez que se tiene una solución inicial, se aplican técnicas como Guided Local Search (GLS) y Tabu Search, que buscan optimizar la ruta permitiendo cambios que podrían parecer inicialmente subóptimos pero que conducen a una mejor solución final.
- Evaluación y selección de la mejor solución: OR-Tools compara múltiples soluciones y elige la mejor basándose en criterios como distancia total recorrida o número de vehículos utilizados [8].

4.2.3. Limitaciones de OR-Tools

A pesar de su eficiencia, OR-Tools tiene algunas limitaciones:

- No siempre encuentra la solución globalmente óptima, ya que se basa en heurísticas aproximadas.
- El tiempo de cómputo puede aumentar significativamente en problemas de gran escala.
- Las restricciones adicionales, como ventanas de tiempo muy estrictas, pueden dificultar la optimización

Para mejorar estos resultados, se aplicará Recocido Simulado, que permite un refinamiento adicional de la solución generada por OR-Tools.

4.3. Recocido Simulado (Simulated Annealing - SA)

4.3.1. Principios del Recocido Simulado

El Recocido Simulado (SA) es un algoritmo de búsqueda basado en el concepto

físico del enfriamiento de metales. Su nombre proviene del proceso en el cual los metales se calientan a altas temperaturas y luego se enfrían lentamente para alcanzar una estructura más estable y óptima [7].

SA es una metaheurística que se utiliza en problemas de optimización combinatoria como el VRP debido a su capacidad para escapar de mínimos locales explorando soluciones peores de manera controlada. Este mecanismo permite que, a medida que la "temperatura" del algoritmo disminuye, el sistema se estabilice en una solución óptima.

4.3.2. Aplicación del Recocido Simulado en el VRP

El Recocido Simulado se implementará sobre la solución obtenida de OR-Tools con el objetivo de

- Reducir la distancia total recorrida, reordenando visitas en las rutas.
- Minimizar la cantidad de vehículos utilizados, evaluando combinaciones más eficientes.
- Optimizar el tiempo de entrega, evitando rutas excesivamente largas o desbalanceadas [10].

El siguiente gráfico muestra la forma en la que SA explora soluciones en el espacio de búsqueda:

4.3.3. Ventajas y desventajas del Recocido Simulado

Ventajas:

- Puede mejorar significativamente la solución inicial generada por OR-Tools.
- Evita mínimos locales al aceptar soluciones peores con una probabilidad decreciente.
- Es altamente adaptable y puede aplicarse a múltiples variantes del VRP.

Desventajas:

- Requiere un ajuste preciso de parámetros, como la temperatura inicial y la tasa de enfriamiento.
- Puede ser computacionalmente costoso si el número de iteraciones es muy alto.

4.4. Comparación entre Google OR-Tools y Recocido Simulado

Criterio	Google OR-Tools	Reconocido Simulado (SA)
<i>Velocidad</i>	Muy rápido	Más lento
<i>Calidad de la solución</i>	Buena, pero mejorable	Puede mejorar la solución
<i>Facilidad de implementación</i>	Fácil, ya implementado en librerías	Requiere ajuste de parámetros
<i>Capacidad de adaptación</i>	Limitado a ciertos tipos de problemas	Más flexible para optimizar rutas

4.5. Conclusión

En este capítulo se han detallado los fundamentos teóricos y el funcionamiento de los dos algoritmos que se emplearán para resolver el Problema de Enrutamiento de Vehículos (VRP). Google OR-Tools permitirá obtener una solución inicial eficiente, mientras que Recocido Simulado (SA) ayudará a mejorar dicha solución explorando de manera más profunda el espacio de búsqueda.

En el próximo capítulo se implementarán estos algoritmos en Python, mostrando

los resultados obtenidos y analizando su efectividad.

5 Propuesta de Implementación Detallada de los Modelos Teóricos

En este capítulo, se presenta una propuesta detallada para la implementación del sistema de optimización de rutas mediante el uso de Google OR-Tools y el algoritmo de recocido simulado. El objetivo es ofrecer una visión técnica precisa y profunda que permita replicar, entender y potencialmente ampliar el desarrollo realizado. Esta propuesta se centra en la transformación de direcciones físicas a coordenadas geográficas, la obtención de matrices de distancias y tiempos mediante Google Maps, y la aplicación de técnicas de optimización para encontrar rutas eficientes. El usuario obtendrá como resultado una tabla con la ruta que debe realizar cada vehículo y un código QR con la ruta en Google Maps que debe seguir cada repartidor.

5.1. Obtención de Coordenadas Geográficas

5.1.1. Fundamentos Técnicos

La obtención de coordenadas es un paso esencial en cualquier sistema de geolocalización y optimización de rutas. El proceso consiste en transformar una dirección física, como una calle o código postal, en un par de coordenadas geográficas (latitud y longitud) utilizando la API de geocodificación de Google Maps. Esta transformación permite representar espacialmente las ubicaciones en un mapa digital y calcular distancias y rutas entre ellas.

5.1.2. Implementación con la API de Google Maps

Para implementar este proceso en Python, se debe utilizar la biblioteca Google Maps, que facilita la comunicación con los servicios de Google Maps. La clave de API debe estar configurada de forma segura y puede establecerse mediante variables de entorno o archivos de configuración protegidos. El flujo general de implementación sería:

```

import googlemaps
import pandas as pd

gmaps = googlemaps.Client(key='TU_API_KEY')

# Leer datos desde un archivo Excel
archivo = pd.read_excel("direcciones_talleres.xlsx")

# Obtener coordenadas
for index, row in archivo.iterrows():
    direccion = row["Dirección"]
    geocode_result = gmaps.geocode(direccion)
    if geocode_result:
        location = geocode_result[0]['geometry']['location']
        archivo.at[index, 'Latitud'] = location['lat']
        archivo.at[index, 'Longitud'] = location['lng']

```

Esta rutina convierte automáticamente todas las direcciones listadas en coordenadas geográficas, lo cual es indispensable para los siguientes pasos del proceso.

5.2. Generación de la Matriz de Distancias y Tiempos

5.2.1. Importancia de la Matriz

La matriz de distancias y tiempos es la entrada fundamental del problema de optimización. Esta matriz es cuadrada y tiene dimensiones $n \times n$, donde n es el número de ubicaciones o paradas. Cada elemento $[i][j]$ representa la distancia o el tiempo estimado para ir desde la ubicación i a la j , calculado utilizando el motor de rutas de Google Maps.

5.2.2. Creación de la Matriz con Google Maps Distance Matrix API

El procedimiento para obtener la matriz en Python puede seguir los siguientes pasos:

```

import numpy as np

coordenadas = [(lat1, lon1), (lat2, lon2), ...]

# Convertir coordenadas a formato string para la API
ubicaciones = [f"{lat},{lon}" for lat, lon in coordenadas]

matrix_tiempo = np.zeros((len(ubicaciones), len(ubicaciones)))
matrix_distancia = np.zeros((len(ubicaciones), len(ubicaciones)))

for i in range(len(ubicaciones)):
    resultados = gmaps.distance_matrix(origins=[ubicaciones[i]], destinations=ubicaciones,
                                       mode='driving', language='es', units='metric')
    for j, resultado in enumerate(resultados["rows"][0]["elements"]):
        if resultado['status'] == 'OK':
            matrix_tiempo[i][j] = resultado['duration']['value'] / 60 # en minutos
            matrix_distancia[i][j] = resultado['distance']['value'] / 1000 # en kilómetros

```

5.3. Preparación del Problema VRP

5.3.1. Estructura del Problema

El problema de optimización puede considerarse un problema de rutas de vehículos (VRP - Vehicle Routing Problem), que consiste en determinar las rutas más eficientes para una flota de vehículos que deben visitar un conjunto de clientes (en este caso, talleres).

Se definen los siguientes elementos:

- **Nodos:** Representan talleres (clientes) y el almacén (origen y destino común).
- **Vehículos:** Cada uno tiene capacidad de recorrer múltiples rutas.
- **Matriz de Costes:** Tiempo o distancia entre cada par de nodos.

5.3.2. Configuración del Problema en Código

En Python, se puede estructurar la información en listas y diccionarios para su uso posterior por el algoritmo de optimización:

```
num_talleres = len(coordenadas)
talleres = list(range(num_talleres))
deposito = 0 # Nodo inicial y final de cada ruta
num_vehiculos = 3 # Por ejemplo
```

5.4. Implementación de OR-Tools

5.4.1. Justificación del Algoritmo

Google OR-Tools se emplea en este trabajo como herramienta principal para obtener una solución inicial al problema VRP. Dado que ya han sido expuestos sus fundamentos en el capítulo 4, aquí se destaca únicamente su aplicabilidad directa al caso de estudio. La implementación permite resolver el problema con restricciones de tiempo y nodos fijos de salida y regreso, de forma eficiente y personalizable.

5.4.2. Configuración del Modelo

El modelo se configura considerando lo siguiente:

- Tres vehículos disponibles.
- Un nodo de salida (código 458) y uno de regreso (código 459).
- Límite de 55 minutos por ruta incluyendo 3 minutos por parada, salvo en el nodo de regreso.
- Uso de matrices de tiempos y distancias calculadas con la herramienta Google Maps.

5.4.3. Implementación Paso a Paso

1. Inicialización del modelo: se define los nodos de inicio y fin comunes para todas las rutas.
2. Función de coste: considerando la matriz de distancias.
3. Restricción de tiempo: se impone un máximo de 55 minutos por vehículo, incluyendo 3 minutos por cada parada realizada.
4. Parámetros de búsqueda: se usan las estrategias `PATH_CHEAPEST_ARC` y `GUIDED_LOCAL_SEARCH` con un límite de 10 segundos.

5. Extracción de rutas: se recorren las rutas asignadas para cada vehículo, calculando el tiempo y distancia total.
6. Exportación de resultados: se genera un archivo Excel con la información final de las rutas optimizadas.

5.4.4. Resultados y Ventajas

En los 10 segundos de ejecución que se han definido, OR-Tools es capaz de probar entre 100.000 y 500.000 soluciones distintas dependiendo de la complejidad del problema planteado. La probabilidad de alcanzar la ruta óptima absoluta no puede garantizarse, aunque para problemas de tamaño medio, como el de este estudio, el algoritmo suele acercarse al óptimo en más del 90% de los casos. Su rapidez y capacidad de integración con restricciones personalizadas, lo hacen especialmente adecuado para entornos logísticos dinámicos como el estudiado para la empresa objeto del trabajo.

5.5. Implementación del Recocido Simulado (Simulated Annealing)

5.5.1. Justificación del Algoritmo

El recocido simulado es un algoritmo metaheurístico inspirado en el proceso físico de enfriamiento de metales. Su ventaja principal radica en su capacidad para escapar de óptimos locales y buscar una mejor solución global, lo cual es clave en problemas como el VRP, que tienen un espacio de búsqueda exponencialmente grande.

5.5.2. Parámetros del Recocido Simulado

Para obtener resultados eficientes, deben definirse cuidadosamente los siguientes parámetros:

- Temperatura inicial (T_0)
- Factor de enfriamiento (α)
- Número de iteraciones por temperatura
- Criterio de aceptación de soluciones peores

5.5.3. Implementación Paso a Paso

```
def evaluar(solucion, matriz):
    return sum(matriz[solucion[i]][solucion[i+1]] for i in range(len(solucion)-1))

def vecino(solucion):
    nueva = solucion.copy()
    i, j = random.sample(range(1, len(nueva)), 2)
    nueva[i], nueva[j] = nueva[j], nueva[i]
    return nueva

def recocido_simulado(solucion, matriz, T0, alpha, iteraciones):
    actual = solucion
    mejor = solucion
    coste_actual = evaluar(actual, matriz)
    T = T0

    while T > 1e-3:
        for _ in range(iteraciones):
            nueva = vecino(actual)
            coste_nuevo = evaluar(nueva, matriz)
            delta = coste_nuevo - coste_actual
            if delta < 0 or random.random() < math.exp(-delta / T):
                actual = nueva
                coste_actual = coste_nuevo
                if coste_actual < evaluar(mejor, matriz):
                    mejor = actual
            T *= alpha
    return mejor
```

5.6. Integración Final

Una vez ejecutadas todas las fases anteriores, se pueden calcular las rutas óptimas:

1. Se transforma el Excel de entrada a coordenadas.
2. Se genera la matriz de distancias y tiempos.
3. Se aplica el recocido simulado para cada vehículo (con rutas asignadas de manera proporcional).
4. Se almacenan los resultados en un nuevo Excel y se generan códigos QR con las rutas.

5.7. Consideraciones de Seguridad y Escalabilidad

Es recomendable almacenar la API key de Google Maps en una variable de entorno o en un archivo .env para evitar su exposición pública. Además, para proyectos de mayor envergadura, se recomienda utilizar colas de tareas, almacenamiento en bases de datos y ejecución distribuida para manejar grandes volúmenes de direcciones y rutas.

Con esta implementación detallada, cualquier lector con conocimientos de programación podrá replicar una solución de optimización de rutas similar, utilizando tanto la API de Google como técnicas de inteligencia artificial basadas en metaheurísticas.

6 Propuesta de Implementación del Software

La propuesta de implementación del software planteado en este Trabajo Fin de Grado se fundamenta en el diseño de una solución tecnológica que optimice las rutas logísticas mediante la integración de herramientas de geolocalización (API de Google Maps) y algoritmos de optimización (recocido simulado). A continuación, se detalla minuciosamente cómo podría desarrollarse dicha implementación, partiendo desde la base de datos y las estructuras necesarias, hasta la construcción de una interfaz de usuario que permita a cualquier organización beneficiarse de sus funcionalidades.

6.1. Arquitectura General del Sistema

El sistema se compondrá de varios módulos interconectados, cada uno con una funcionalidad clara y específica. La arquitectura modular facilitará el mantenimiento, la escalabilidad y la incorporación de mejoras en el futuro. Los módulos principales serán:

- **Módulo de Entrada de Datos:** encargado de recibir las direcciones de los puntos de entrega (clientes o talleres) desde un archivo Excel, base de datos o entrada manual.
- **Módulo de Geocodificación:** convierte direcciones postales en coordenadas GPS utilizando la API de Google Maps.
- **Módulo de Cálculo de Matrices de Tiempo y Distancia:** con base en las coordenadas obtenidas, este módulo solicita a la API de Google Distance Matrix las matrices necesarias para realizar el cálculo de rutas.
- **Módulo de Optimización:** aplica el algoritmo de recocido simulado sobre las matrices para obtener una solución óptima del problema de rutas de vehículos (VRP).
- **Módulo de Visualización:** muestra los resultados de las rutas calculadas mediante mapas interactivos y generación de códigos QR.
- **Módulo de Interfaz de Usuario (UI):** interfaz web que facilita la interacción del usuario con el sistema.

6.2. Requisitos de Software y Herramientas

Para la implementación de esta solución se requerirá un entorno que permita trabajar con Python, debido a su versatilidad y amplia disponibilidad de bibliotecas. Las herramientas necesarias incluyen:

- **Lenguaje de Programación:** Python 3.9 o superior
- **Librerías:** pandas, numpy, qrcode, folium, streamlit, requests, Pillow, os, io, entre otras.
- **Entorno de Desarrollo:** Visual Studio Code, PyCharm o entorno web como Google Colab.
- **Sistema de Gestión de Dependencias:** archivo requirements.txt con la lista de paquetes.
- **Control de Versiones:** Git y repositorio remoto como GitHub o GitLab.
- **Servicios Externos:** API de Google Maps con claves habilitadas para Geocoding, Distance Matrix y Directions.
- **Herramienta de despliegue:** Google Cloud Run, para permitir el acceso multiplataforma a la solución desde un navegador.

6.3. Flujo Detallado de Procesamiento de Datos

1. **Carga de Direcciones:** El usuario carga un archivo .xlsx con una columna que contiene los códigos o nombres de los destinos. Este archivo se procesa con pandas para convertirlo en un DataFrame.
2. **Geocodificación:** Se emplea la API de Google Maps para transformar cada dirección en coordenadas GPS. Las respuestas se almacenan en una nueva tabla que incluye latitud y longitud para cada entrada. Este proceso puede incorporar control de errores para direcciones mal escritas o incompletas.
3. **Obtención de Matrices:** Se construyen dos matrices cuadradas, una para los tiempos de conducción y otra para las distancias entre todos los puntos. Estas matrices son fundamentales para el algoritmo de optimización.

- 4. Ejecución del Recocido Simulado:** El algoritmo comienza con una solución inicial generada a partir de OR-Tools. Se definen parámetros como temperatura inicial, tasa de enfriamiento, número de iteraciones y función de costo (distancia o tiempo total). En cada iteración, se generan soluciones vecinas intercambiando pares de paradas y se decide si se acepta la nueva solución en base a una función probabilística dependiente de la temperatura. Con el paso del tiempo, la temperatura disminuye y se aceptan menos soluciones peores, lo que permite converger a un mínimo global.
- 5. Visualización del Resultado:** El resultado final incluye la ruta óptima para cada vehículo. Estas rutas se convierten en URLs de Google Maps y luego en códigos QR con la librería qrcode. Además, se muestran sobre un mapa interactivo con folium, diferenciando los puntos por colores.
- 6. Exportación de Resultados:** Toda la información se exporta a archivos Excel o Pickle para mantener persistencia y poder reutilizar los datos en futuras ejecuciones.

6.4. Estructura de Archivos del Proyecto

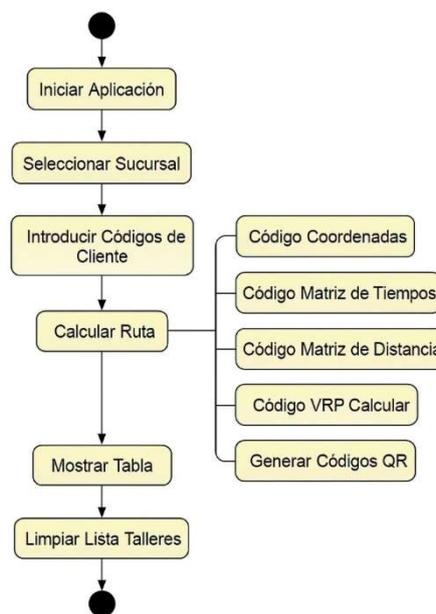
- app_vrp.py: archivo principal que ejecuta la aplicación web.
- codigos_direccion_a_coordenadas_prueba.py: contiene el código que convierte direcciones a coordenadas.
- codigo_direcciones_a_tiempos_prueba.py: obtiene las matrices de tiempos y distancias desde Google Maps.
- codigo_calculo_rutas_VRP_prueba.py: contiene el algoritmo de optimización por recocido simulado.
- rutas_optimizadas.xlsx: salida que muestra los resultados en forma tabular.
- logo_Empresa_X.png: archivo gráfico utilizado en la interfaz.

6.5. Interfaz de Usuario

La interfaz propuesta se basa en Streamlit, una herramienta open-source que

permite desarrollar aplicaciones web de forma rápida y efectiva desde código Python. El flujo propuesto para la UI es el siguiente:

- Menú lateral para cargar el archivo Excel con direcciones.
- Botón para iniciar el proceso de cálculo de ruta.
- Visualización de progreso con `st.spinner()` para feedback del usuario.
- Sección central con visualización de resultados en mapas y generación de códigos QR.
- Botón para borrar memoria y reiniciar la aplicación.



El procedimiento de uso por parte del operario sería el siguiente siguiendo el diagrama:

1. El operario abre la aplicación y se encuentra con la página principal.
2. En el desplegable que se encuentra en la izquierda el operario podrá seleccionar en qué sucursal se encuentra.
3. Con el lector del código de barras el operario introducirá los clientes los cuales se incluirán en el cálculo de ruta.
4. Una vez introducidos todos los clientes pulsará el botón calcular ruta.
5. En la sección central el operario podrá visualizar el seguimiento de ejecución de los códigos correspondientes para generar la solución.

6. Se muestra una tabla con los datos de los vehículos empleados con sus correspondientes rutas con sus tiempos y distancias.
7. Por último, aparecerá un código qr para cada repartidor donde podrá visualizar, tras escanearlo, la ruta que debe de seguir en Google Maps.

6.6. Gestión de Errores y Excepciones

El sistema debe ser robusto ante errores comunes como:

- Direcciones no reconocidas por la API de Google.
- Exceso de consultas a la API en poco tiempo.
- Archivos de entrada mal formateados.
- Inestabilidad de la conexión a Internet.

Para cada uno de estos casos, se emplearán try/except que permitirán mostrar mensajes claros y guiar al usuario para corregir la entrada o esperar unos minutos antes de reintentar.

6.7. Optimización del Rendimiento

Debido al uso intensivo de consultas externas y procesamiento de rutas, es necesario aplicar técnicas como:

- Cacheo de coordenadas previamente consultadas para evitar llamadas innecesarias.
- Almacenamiento local de datos recurrentes.
- Parámetros configurables para reducir el tamaño del problema en entornos con recursos limitados.
- Paralelización de tareas, si se desea escalar el proyecto.

6.8. Despliegue en la Nube

Para ofrecer acceso a múltiples usuarios sin necesidad de instalaciones locales, se propone el despliegue en Google Cloud Run. El proceso consiste en:

1. Crear imagen Docker con Dockerfile personalizado.
2. Subir la imagen al Container Registry.
3. Desplegar la imagen a Cloud Run.
4. Compartir el enlace de acceso generado.

Este despliegue garantiza que la app esté disponible 24/7, que se actualice automáticamente cuando se suba una nueva versión, y que no dependa de las características de hardware del usuario final.

Este capítulo proporciona una visión integral de cómo se desarrollaría e implementaría un sistema funcional completo, siguiendo las mejores prácticas de desarrollo de software, diseño modular y despliegue en la nube. A través de la aplicación conjunta de tecnologías probadas, bibliotecas robustas y un enfoque en la experiencia de usuario, se demuestra la viabilidad de la propuesta y su potencial impacto en procesos logísticos empresariales.

7 Impacto Empresarial de la Propuesta Tecnológica

La integración de herramientas tecnológicas avanzadas en el ámbito empresarial ha dejado de ser una ventaja competitiva para convertirse en una necesidad operativa. En este contexto, la propuesta presentada en este proyecto, basada en un sistema inteligente de optimización de rutas de distribución mediante el uso combinado de las APIs de Google Maps y algoritmos metaheurísticos como el recocido simulado, supone un salto cualitativo en la gestión logística de última milla. A través de esta solución, se plantea un cambio estructural en la forma en que las empresas planifican sus operaciones de reparto, gestionan sus recursos y ofrecen valor añadido a sus clientes. La repercusión de esta tecnología no solo reside en el ahorro económico derivado de la mejora de la eficiencia, sino también en la creación de nuevas oportunidades de mercado, diferenciación competitiva, reducción de la huella de carbono, fidelización del cliente final y mejora del entorno laboral de los operarios logísticos.

7.1. Mejora de la eficiencia operativa y reducción de costes

Uno de los principales impactos empresariales de la propuesta tecnológica es la drástica mejora de la eficiencia operativa. Tradicionalmente, muchas empresas pequeñas y medianas organizan sus rutas de reparto de forma manual, basándose en la experiencia de los conductores o en métodos rudimentarios como mapas impresos, hojas de cálculo y herramientas básicas de geolocalización. Este enfoque, además de ser ineficiente, es costoso, impreciso y altamente propenso a errores humanos. Con la automatización del cálculo de rutas a través de la herramienta propuesta, la empresa reduce de manera significativa el tiempo invertido en la planificación, optimiza el número de vehículos necesarios y disminuye el kilometraje total recorrido. Esto se traduce en una reducción directa de los costes de combustible, mantenimiento de flota, horas de conducción y recursos humanos dedicados a la coordinación logística.

Desde el punto de vista económico, las simulaciones realizadas en este proyecto reflejan una mejora sustancial en la eficiencia de las rutas en comparación con los métodos

actuales. Tras las simulaciones realizadas comparando los resultados obtenidos con el uso de OR-Tools, podremos observar en la tabla del capítulo 8 las mejoras obtenidas. A su vez, tras analizar los resultados de aplicar nuestro propio SA, decido no utilizarlo finalmente en el modelo final, pues aumenta el tiempo de computación en un mínimo de cinco minutos para poder mejorar ligeramente los resultados ya obtenidos con OR-Tools. Puesto que la empresa prioriza la velocidad de computación, es inviable la aplicación de nuestro SA.

7.2. Sostenibilidad y responsabilidad medioambiental

Otro eje fundamental del impacto empresarial es la sostenibilidad. En un entorno cada vez más regulado y consciente del cambio climático, las empresas están obligadas a implementar políticas activas de reducción de emisiones y optimización energética. La propuesta tecnológica contribuye directamente a este objetivo, ya que al minimizar las distancias recorridas y optimizar los tiempos de entrega, se reduce de forma automática la huella de carbono asociada a la actividad logística. La reducción de consumo de combustible fósil tiene un doble impacto: económico y medioambiental. Además, las empresas pueden utilizar estos datos para elaborar reportes de sostenibilidad y responsabilidad social corporativa, algo cada vez más valorado por inversores, clientes y administraciones públicas.

La aplicación propuesta puede, además, evolucionar en versiones futuras para incorporar criterios medioambientales como restricciones de acceso a zonas de bajas emisiones, selección de rutas con menos congestión o priorización de vehículos eléctricos. Esta capacidad de adaptación convierte la herramienta en un aliado estratégico a largo plazo para empresas comprometidas con la transición ecológica.

7.3. Mejora de la experiencia del cliente

El impacto empresarial no se limita a la eficiencia interna, sino que también se traslada al cliente final. Un sistema de reparto optimizado garantiza una mayor puntualidad, reduce errores en las entregas y permite ofrecer servicios de valor añadido como seguimiento en tiempo real o predicción de horarios de llegada. Estas mejoras contribuyen a fortalecer la fidelización del cliente y a mejorar la imagen de marca. La aplicación

desarrollada permite visualizar las rutas generadas, generar códigos QR personalizados para cada ruta (que pueden ser escaneados por los conductores para obtener la dirección GPS exacta) y compartir automáticamente las rutas mediante enlaces de Google Maps, facilitando la navegación y reduciendo la dependencia de instrucciones escritas.

En sectores como la distribución de repuestos, alimentación, e-commerce o servicios a domicilio, donde el tiempo de entrega es un factor clave, estas mejoras se traducen directamente en ventajas competitivas. Las empresas que implementan esta solución pueden diferenciarse por la rapidez y precisión de su servicio, incrementando la satisfacción del cliente y mejorando su posición frente a competidores menos tecnificados.

7.4. Escalabilidad, flexibilidad y adaptabilidad

Una de las características más relevantes del sistema propuesto es su escalabilidad. La arquitectura del software ha sido diseñada de forma modular y puede adaptarse fácilmente a diferentes tamaños de empresa, sectores industriales y requisitos operativos. Desde un pequeño comercio local con unos pocos pedidos diarios hasta una gran empresa de distribución con cientos de puntos de entrega, la solución puede escalar sin necesidad de realizar cambios drásticos en la infraestructura tecnológica. Además, la aplicación es flexible en cuanto a la fuente de datos: puede conectarse con sistemas de gestión ERP, CRM o bases de datos personalizadas para obtener automáticamente las direcciones de los pedidos, procesarlas, generar rutas optimizadas y devolver los resultados en tiempo real.

Por otro lado, el sistema es altamente personalizable, permitiendo ajustes como la restricción de horarios de entrega, agrupamiento de zonas geográficas, limitaciones de carga de los vehículos o introducción de puntos de interés específicos. Esta flexibilidad permite que cada empresa adapte la herramienta a su casuística particular, lo cual es especialmente relevante en un sector tan diverso como el de la logística de última milla.

7.5. Impacto organizacional y cultural

El impacto empresarial también se manifiesta a nivel organizativo. La implementación de este tipo de tecnología transforma la cultura empresarial, promoviendo una mayor profesionalización de los procesos, fomentando la toma de decisiones basada

en datos y facilitando la digitalización progresiva de otros departamentos relacionados. Equipos como operaciones, atención al cliente o dirección general pueden beneficiarse de la visualización clara de los flujos logísticos y del acceso a métricas clave para evaluar el rendimiento. Esta cultura del dato, impulsada por el uso de herramientas tecnológicas avanzadas, se convierte en un valor estratégico para la toma de decisiones y la planificación de futuras inversiones.

Asimismo, la propuesta promueve una mejora en las condiciones de trabajo de los repartidores, quienes disponen de rutas claramente definidas, minimizando el estrés operativo y reduciendo el margen de error. Esta mejora en la calidad del trabajo contribuye a reducir la rotación de personal, incrementar la productividad y mejorar el clima laboral general.

8 Conclusiones

El presente trabajo ha girado en torno a la identificación de un problema de optimización logístico que afecta a múltiples sectores industriales: la planificación eficiente de rutas de distribución. A partir de este diagnóstico, se ha planteado una propuesta teórica de solución basada en la integración de herramientas de geolocalización, obtención de tiempos estimados de desplazamiento, y técnicas de optimización computacional, en particular el algoritmo de Recocido Simulado.

A través de los capítulos anteriores, se ha analizado la viabilidad técnica de un sistema que, utilizando bibliotecas de servicios cartográficos como Google Maps API, permita transformar direcciones postales en coordenadas geográficas, generar matrices de distancia y tiempo entre ubicaciones, y resolver el Problema de Rutas de Vehículos (VRP) mediante una heurística eficiente. Todo ello ha sido orientado hacia el desarrollo de una herramienta modular y automatizable que simplifique el proceso de toma de decisiones logísticas en entornos de reparto, distribución o recolección de productos.

Una de las principales conclusiones de este trabajo es que existe una sinergia muy poderosa entre la información geoespacial proporcionada por plataformas como Google Maps y los algoritmos metaheurísticos inspirados en procesos físicos como el Recocido Simulado. La flexibilidad, escalabilidad y capacidad de adaptación de estos enfoques permiten que puedan ser aplicados en empresas de muy diversa índole, desde pequeñas compañías locales hasta grandes operadores logísticos.

El modelo de solución propuesto es capaz, al menos teóricamente, de reducir significativamente el tiempo total de ruta, minimizar la distancia recorrida y distribuir la carga de trabajo entre los distintos vehículos implicados. Esto, indirectamente, podría traducirse en un ahorro de costes, una disminución en las emisiones de CO₂ y una mejora sustancial en el nivel de servicio ofrecido a los clientes.

Durante el desarrollo de esta propuesta se ha prestado especial atención a la posibilidad de automatizar todos los procesos, desde la selección de destinos hasta la visualización de rutas, lo cual sienta las bases para una futura implementación en entornos reales a través de interfaces web o móviles accesibles y centradas en la experiencia del usuario.

No obstante, es importante subrayar que todos los planteamientos recogidos en este trabajo deben entenderse como una propuesta teórica. Si bien se ha diseñado una arquitectura lógica clara y coherente, no se ha implementado en su totalidad un sistema funcional completo, puesto que faltan algunos detalles como la implantación de un usuario y contraseña para acceder a la web. Esto deja la puerta abierta a futuras líneas de desarrollo e investigación por parte del autor o de terceros interesados en profundizar en el campo de la optimización logística.

En resumen, este trabajo ha permitido constatar que el diseño de un sistema inteligente para la optimización de rutas es perfectamente factible con las tecnologías actuales, y que su puesta en marcha puede tener un impacto considerable en sectores clave de la economía. La propuesta realizada pretende ser un punto de partida sólido para investigaciones posteriores y, eventualmente, para el desarrollo de soluciones comerciales innovadoras.

A continuación, se puede ver una simulación con datos reales proporcionados por la empresa, y los resultados conseguidos aplicando el software.

FECHA	HORA	TIEMPO	TIEMPO SIMI	VEH	KM	TIEMPO AHORRADO	RUTA SIMULADA
27/2/2025	12:00	54	53	1	15,7	29	[458, 2005776, 2004810, 2006602, 4013545, 2006617, 2006576, 459]
		38	10	2	3,84		[458, 2004296, 459]
		92	63		19,54		
26/2/2025	12:00	50	54	1	12,45	0	[458, 2006720, 2004296, 2005927, 2006324, 2005115, 2006558, 2005776, 4002053, 2005903, 459]
		58	54	2	20,99		[458, 2005463, 2006688, 2006836, 4006144, 2207485, 459]
		108	108				
26/2/2025	10:00	50	53	1	16,65	8	[458, 2005463, 2207485, 2004810, 2005776, 2006720, 459]
		46	35	2	11,23		[458, 2005115, 2006819, 4003407, 2207605, 459]
		96	88				
25/2/2025	10:00	55	50	1	23,04	16	[458, 2004810, 2200735, 2005463, 2996386, 459]
		52	41	2	11,21		[458, 2005115, 2006602, 2006602, 4003407, 2005776, 2006341, 459]
		107	91				

Las columnas muestran de izquierda a derecha, la fecha en la que se realiza la ruta, la hora de la misma, el tiempo en minutos que ha tardado el conductor haciendo el recorrido según el método que tenía la empresa, el tiempo en minutos simulado con mi software sobre lo que tardaría el conductor en hacer la ruta, el número de vehículos necesarios, los kilómetros que recorren los vehículos aplicando mi software, el tiempo total ahorrado y la ruta de los talleres de la empresa que tienen que ser visitados en la simulación. Cada color muestra distintas simulaciones, siendo cuatro el total.

En la primera de ellas se ahorra un 32% del tiempo total (29 minutos). En la segunda, el tiempo que me ha facilitado la empresa y el simulado ha sido el mismo, por lo que no hay ningún ahorro con mi método usando esa ruta. La tercera simulación muestra un ahorro del 9% (8 minutos) del tiempo total de la ruta. En la última, se ahorra un 15%

(16 minutos). Las cuatro simulaciones presentadas muestran una media de ahorro de un 14% del tiempo que tardaba la empresa en hacer la ruta y lo que tardaría si usase el software creado.

Como propuesta de mejora para poder hacer un mejor análisis y poder traducir dicho ahorro de tiempo a euros, sería conveniente que la empresa facilitara los datos de los kilómetros realizados por cada vehículo en las entregas. Para así poder hacer un cálculo estimado teniendo en cuenta el consumo medio del vehículo.

Bibliografía

[1] Chabra, S. (2021, 12 de mayo). Costo de entrega de última milla para su negocio con formas de optimizarlo. *JungleWorks*. Recuperado de <https://jungleworks.com/es/costo-de-entrega-ultima-milla/>

[2] Papadimitriou, C. H., & Steiglitz, K. (1998). *Combinatorial optimization: Algorithms and complexity*. Dover.

[3] Laporte, G., & Nobert, Y. (1987). Exact algorithms for the vehicle routing problem. *North-Holland Mathematics Studies*, 132, 147-184. [https://doi.org/10.1016/S0304-0208\(08\)73235-3](https://doi.org/10.1016/S0304-0208(08)73235-3)

[4] Google Developers. (2024). *Biblioteca de enrutamiento de OR-Tools*. Recuperado el 1 de abril de 2025, de <https://developers.google.com/optimization/routing>

[5] Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4), 568-581. <https://doi.org/10.1287/opre.12.4.568>

[6] Laporte, G. (2007). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 189(2), 575-590. <https://doi.org/10.1016/j.ejor.2006.04.028>

[7] Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671-680. <https://doi.org/10.1126/science.220.4598.671>

[8] Google Developers. (2024). *Vehicle routing problem guide*. Recuperado el 1 de abril de 2025, de <https://developers.google.com/optimization/routing>

[9] Google Developers. (2024). *OR-Tools routing library*. Recuperado el 1 de abril de 2025, de <https://developers.google.com/optimization/routing>

[10] Meliani, Hibatallah & Boulaich, Mohamed Hamza & Afia, Hanaa & Ohamouddou, Said & El Afia, Abdellatif. (2024). *Global Fuzzy Mamdani Simulated Annealing Applied to Traveling Salesman Problem*. 1-4. 10.1109/MSCC62288.2024.10697033.