

Sistema de Optimización Multi Algorítmica para Procesos de Distribución Inteligente

Iker Acevedo Vargas

Universidad Sergio Arboleda  
Escuela de Ciencias Exactas e Ingeniería

Javier Ochoa

SIST 0175 – Análisis de Algoritmos

Bogotá D.C, Noviembre 2025

## 1. Introducción:

El presente documento analiza los fundamentos conceptuales y técnicos necesarios para el desarrollo de un sistema de optimización logística inteligente, diseñado para mejorar la gestión de pedidos, rutas y recursos dentro de un entorno de distribución.

El sistema permitirá ordenar pedidos según prioridad, calcular rutas de entrega óptimas y asignar vehículos o repartidores de forma eficiente, aplicando distintos algoritmos clásicos y heurísticos.

El objetivo es comprender cómo las estrategias algorítmicas influyen en el rendimiento, la eficiencia y la escalabilidad de un sistema logístico, comparando su comportamiento teórico y empírico mediante pruebas con datasets simulados.

## 2. Definición del problema:

Para este proyecto se implementará un sistema de distribución y entrega de pedidos, donde esté enfrente desafíos constantes relacionados con:

- La priorización de pedidos según su urgencia o ubicación
- La optimización de rutas para minimizar tiempo y distancia
- La asignación eficiente de trabajos para cada parte de la empresa

Una mala gestión algorítmica puede generar costos elevados, retrasos y bajo rendimiento computacional.

Por ello, este proyecto busca modelar una solución capaz de ordenar, calcular y asignar utilizando algoritmos de diferentes paradigmas, analizando su complejidad temporal, en memoria y desempeño empírico.

## 3. Objetivos

Objetivo general

Desarrollar un sistema logístico que implemente y compare algoritmos de ordenamiento, rutas y asignación de recursos, evaluando su rendimiento teórico y empírico sobre datasets simulados

Objetivos específicos:

- Analizar el problema de distribución y definir su modelo computacional.
- Implementar algoritmos de ordenamiento para priorizar pedidos
- Aplicar algoritmos de rutas óptimas para hallar los mejores trayectos
- Utilizar el Algoritmo Húngaro para asignar recursos con costos mínimos
- Medir y comparar los tiempos de ejecución y complejidades
- Generar reportes visuales y conclusiones basadas en datos

#### 4. Justificación

La optimización algorítmica es una habilidad esencial para potencializar sistemas. La realización de este proyecto me permitirá:

- Aplicar distintos algoritmos para solucionar problemas
- Visualizar el impacto al implementar estos algoritmos y la mejora en problemas logísticos
- Fortalecer competencias técnicas en diseño, análisis y validación empírica.

Además el sistema podrá escalar a contextos de IA aplicada, enrutamiento de vehículos o planificación de recursos muy demandados en la industria tecnológica actual

#### 5. Descripción de cada algoritmo

- Submódulo 1 - Ordenamiento de pedidos

Algoritmo principal:

Quicksort va a ser el algoritmo principal debido a que este permite organizar arreglos con mucha rapidez y poco consumo de memoria, la complejidad promedio de este algoritmo está en  $O(n \log n)$ , para tener en cuenta en este algoritmo debemos usar un buen pivote ya que si no lo aplicamos bien podemos tener problemas en tema de eficiencia del sistema.

Algoritmo secundario:

MergeSort será el algoritmo secundario debido a que este sigue el paradigma de QuickSort y una complejidad similar, este tiene un buen rendimiento en pequeños y grandes datos, la diferencia principal será su uso de memoria lo cual será uno de los puntos importantes para comparar entre estos dos algoritmos.

Otro algoritmo que se podía usar era HeapSort solo que este cambia su tipo de paradigma y su rendimiento es bastante inferior comparado con QuickSort.

- Submódulo 2 – Rutas de entrega óptimas

Algoritmo principal:

Dijkstra será el algoritmo principal debido a que este tiene una programación dinámica  $O(V^2)$  o  $O(E \log V)$ , la ventaja de utilizar este algoritmo es su precisión en encontrar la ruta más corta sin embargo al probar con grafos muy grandes este pierde precisión y eficiencia. Para este proyecto se utilizará rutas fijas conocidas.

Algoritmo secundario:

A\* será el algoritmo secundario este tiene un tipo de paradigma Heurístico su complejidad depende de heurística  $O(E)$ , la ventaja de usar este algoritmo es su facilidad para implementarse en mapas reales, para este algoritmo se

diseñará una función que permita que el algoritmo funcione de la mejor manera. Este será utilizado en rutas dinámicas o con estimación

- Submódulo 3 – Asignación de recursos

Algoritmo principal:

Algoritmo Húngaro será el algoritmo principal para la asignación de recursos.

Este se basa en el paradigma de optimización combinatoria y tiene una complejidad de  $O(n^3)$ . Su principal ventaja es que encuentra la asignación óptima con el costo mínimo, lo que lo hace ideal para sistemas donde la precisión y eficiencia son críticas.

Sin embargo, su complejidad puede ser un problema en matrices muy grandes, por lo que se debe evaluar el tamaño de los datos antes de implementarlo.

Este algoritmo es especialmente útil en aplicaciones como la asignación de vehículos a rutas o pedidos, donde se requiere una solución óptima.

Algoritmo secundario:

Greedy (alternativo) será el algoritmo secundario. Este sigue un enfoque avaro o heurístico, con una complejidad de  $O(n^2)$ . Su principal ventaja es que es fácil de implementar y rápido, lo que lo hace útil en escenarios donde se necesita una solución rápida aunque no necesariamente óptima.

La desventaja es que puede no encontrar la mejor asignación posible, por lo que se debe usar con precaución en sistemas donde la precisión es importante.

Se recomienda para asignaciones rápidas no exactas, como decisiones en tiempo real o sistemas con recursos limitados.

6. Herramientas y datos a utilizar

Elemento	Herramienta / Librería	Uso previsto
Lenguaje de	Python 3.12	Implementación principal

programación		
Control de versiones	GitHub	Historial y colaboración
Análisis de datos	NumPy, Pandas	Lectura y manejo de datasets
Visualización	Matplotlib, Seaborn	Generación de gráficas de rendimiento.
Modelado de grafos	NetworkX	Implementación de Dijkstra y A*
Dataset de pedidos	pedidos.csv	ID, prioridad, destino, peso
Dataset de rutas	rutas.csv	Nodos, distancias, conexiones
Dataset de recursos	asignacion.csv	Vehículos, rutas, costos

## 7. Casos de uso del sistema

- Ordenar pedidos: Prioriza pedidos según distancia o urgencia.
- Calcular ruta: Determina el camino más corto entre dos puntos
- Asignar recursos: Asigna vehículos a rutas minimizando costo
- Comparar eficiencia: Evalúa tiempos y complejidades de los algoritmos

## 8. Alcance y resultados esperados

Se espera desarrollar un sistema funcional que integre al menos tres algoritmos de optimización aplicados a un contexto logístico real. El proyecto permitirá analizar y comparar el rendimiento teórico y empírico de los algoritmos implementados mediante tablas y gráficas de tiempo y complejidad. Además, se generará un informe final con los resultados obtenidos y un repositorio en GitHub que contenga el código fuente documentado, garantizando la trazabilidad y replicabilidad del proceso de desarrollo.