



**TECNOLÓGICO  
NACIONAL DE MÉXICO®**



**TECNOLOGICO NACIONAL DE MEXICO.  
INSTITUTO TECNOLÓGICO DE CULIACÁN.**

***OPTIMIZACIÓN DE RUTAS DE DISTRIBUCIÓN LOGÍSTICA USANDO EL  
ALGORITMO DE RECOCIDO SIMULADO***

**TOPICOS DE INTELIGENCIA ARTIFICIAL 12-13.**

**DOCENTE:** MORA FELIX ZURIEL DATHAN.

**CLAVE DE LA MATERIA:** GTD2102

**NOMBRE DE LOS ALUMNOS:**

- BORBÓN SÁNCHEZ EDGAR - 16171301
- MILLÁN LÓPEZ ANA KAREN - 20170985

**UNIDAD:** 2

**ACTIVIDAD:** PROYECTO DE OPTIMIZACIÓN DE RUTAS CON RECOCIDO SIMULADO.

**CULIACÁN, SINALOA.**

## Contenido

Introducción.....	3
Objetivos .....	3
Objetivo General.....	3
Objetivos Específicos .....	3
Justificación y Alcance .....	4
Justificación .....	4
Alcance .....	4
Documentos y Organización del Informe .....	4
Desarrollo .....	5
Modelado del Problema .....	5
Algoritmo: Recocido Simulado.....	5
Estrategia de Vecindad y Parámetros.....	5
Implementación Geoespacial y Visualización .....	6
Pruebas Realizadas: Análisis de Parámetros .....	6
Prueba 1 (Búsqueda Exhaustiva).....	6
Prueba 2 (Búsqueda Restringida) .....	7
Conclusiones.....	10
Referencias .....	11
Link – Repositorio.....	11

## Introducción

El Problema de **Ruteo de Vehículos (VRP)** es un desafío central en la logística, crucial para reducir costos operativos y mejorar la eficiencia en la cadena de suministro. Este proyecto aborda la problemática específica de la distribución de productos comestibles desde centros de distribución hacia sucursales de una cadena de restaurantes en Culiacán, Sinaloa. La complejidad del problema, dada la naturaleza combinatoria de las posibles rutas, lo convierte en un candidato ideal para ser resuelto mediante métodos de **búsqueda heurísticos** (Applegate, Bixby, Chvátal, & Cook, 2011).

El objetivo principal es diseñar una solución computacional que determine las rutas óptimas, enfocándose en la minimización del costo total, el cual incluye distancia y combustible, utilizando el algoritmo de **Recocido Simulado**.

## Objetivos

### Objetivo General

Diseñar e implementar una solución computacional basada en el algoritmo heurístico de **Recocido Simulado** para optimizar las rutas de distribución, mejorando la eficiencia logística y reduciendo costos operativos dentro de la zona de estudio.

### Objetivos Específicos

1. Analizar el problema logístico de distribución de productos comestibles en la zona de estudio, modelándolo como un **Problema de Ruteo de Vehículos con Múltiples Depósitos (MDVRP)** (Rodríguez-Vásquez, 2019).
2. Seleccionar y justificar el uso del Recocido Simulado como el algoritmo heurístico más adecuado.
3. Desarrollar una implementación computacional utilizando el lenguaje **Python** y sus librerías especializadas (Kaggle, s.f.).
4. Evaluar el desempeño de la solución mediante métricas de costo total y distancia.

## Justificación y Alcance

### Justificación

La resolución del problema mediante un método de búsqueda heurístico, como el Recocido Simulado, es fundamental ya que los métodos de optimización exactos no son viables en un entorno real con un gran número de nodos. El uso de Recocido Simulado permite explorar soluciones subóptimas y **escapar de mínimos locales**, lo que es esencial para problemas NP-difíciles (Osman, 1993).

La mejora en la toma de decisiones logísticas impacta directamente en la rentabilidad de la empresa al reducir los gastos de combustible (Álvarez Martínez, Toro Ocampo, & Gallego Rendón, 2008).

### Alcance

El proyecto se centra en la optimización de la **asignación de tiendas a Centros de Distribución** y la **secuencia de entrega** dentro de cada ruta. El costo de la ruta se basa en una matriz ponderada de **distancia física y consumo de combustible**, definida en la función **generarMatrizCosto** (Ver [recocido.py](#)).

## Documentos y Organización del Informe

La organización del informe y la implementación de la solución se basaron en el análisis de los siguientes documentos:

Documento	Fuente en el Código	Propósito en el Modelo
Tabla de distribución de la tienda	<a href="#"><u>datos_distribucion_tiemdas.xlsx</u></a>	Proporciona coordenadas geoespaciales y la clasificación de cada nodo (CD/Tienda). Base para la visualización.
Matriz de distancia	<a href="#"><u>matriz_distancias.csv</u></a>	Determina la distancia real (km) entre cada par de nodos.
Matriz de costos de combustible	<a href="#"><u>matriz_costos_combustible.csv</u></a>	Se utiliza para modelar el consumo de combustible entre nodos, integrándose a la matriz de costo final.
Mapa geoespacial actualizado	Implementado con GeoPandas y Contextily ( <a href="#"><u>map.py</u></a> )	Permite la visualización y el análisis de las rutas óptimas generadas.

## Desarrollo

### Modelado del Problema

El sistema modela el **MDVRP** donde cada tienda debe ser asignada y visitada por un vehículo que inicia y termina su recorrido en el mismo CD (Rodríguez-Vásquez, 2019).

**Función de Costo:** La función calcularCostoSolucion evalúa la calidad de cada estado o solución vecina. Esta función no solo suma el costo operativo de la ruta (distancia + combustible), sino que incluye una **Penalidad Geográfica** (PENALIDAD\_GEOGRAFICA = 20000):

$$\text{Costo Total} = \sum_{\text{Rutas}} \text{Costo Operativo} + (\text{Penalidad} \times \text{Tiendas mal asignadas})$$

La penalización es esencial para obligar al algoritmo a converger a una **asignación geográficamente lógica** (tienda al centro más cercano), a medida que la temperatura desciende.

### Algoritmo: Recocido Simulado

El Recocido Simulado (SA) se implementa en la función **recocido**. El criterio de aceptación sigue la distribución de Boltzmann:

$$P = e^{-\frac{\Delta \text{Costo}}{T}}$$

Donde  $\Delta \text{Costo}$  es la diferencia de costo entre el vecino y la solución actual, y  $T$  es la temperatura actual del sistema. La probabilidad  $P$  permite que soluciones peores ( $\Delta \text{Costo} > 0$ ) sean aceptadas con mayor frecuencia a altas temperaturas (exploración).

### Estrategia de Vecindad y Parámetros

El éxito del algoritmo depende de la calibración de los parámetros y de la estrategia de movimiento (operador de vecindad).

Parámetro / Estrategia	Valor/Tipo	Justificación para la Elección
Temperatura Inicial ( $T_{inicial}$ )	2000	Suficientemente alta para permitir una <b>exploración robusta</b> al inicio y evitar óptimos locales.

Tasa de Enfriamiento ( $\alpha$ )	0.995	Implementa un <b>enfriamiento lento</b> , necesario para que el algoritmo explore con profundidad
Operador de Vecindad	Swap (80%) y Reasignación (20%)	El <b>80%</b> de <b>Swap</b> (intercambio) optimiza el orden de visita (explotación). El <b>20%</b> de <b>Reasignación</b> mueve una tienda a un Centro Permitido (exploración a nivel de asignación).
Iteraciones	20,000	Número necesario para que el esquema de enfriamiento lento ( $\alpha=0.995$ ) tenga tiempo de converger.

## Implementación Geoespacial y Visualización

La visualización es manejada por el módulo [map.py](#).

- **Procesamiento de Datos:** Los datos de latitud/longitud se convierten a objetos **Point** de Shapely y se agrupan en un GeoDataFrame (**gdf**).
- **Conversión de Proyección:** Las coordenadas originales (**EPSG:4326**) se transforman al sistema **Web Mercator (EPSG:3857)**, lo cual es crucial para que los puntos se superpongan correctamente sobre el mapa base de OpenStreetMap, añadido mediante la librería **Contextily**.
- **Visualización de Rutas:** La función **crearMapaDeRutas** traza las rutas óptimas (objetos **LineString** de Shapely) y etiqueta cada nodo como "T#" (Tienda) o "CD#" (Centro de Distribución).

## Pruebas Realizadas: Análisis de Parámetros

### Prueba 1 (Búsqueda Exhaustiva)

- **Parámetros:** Tinicial = 2000, 20,000 Iteraciones.
- **Mejor Costo Encontrado:** \$1,074.27.

```
--- Generando solución inicial ---

Costo de la solución inicial: 282,494.62
Iteración 2000/20000 Mejor costo hasta ahora: 41,273.08
Iteración 4000/20000 Mejor costo hasta ahora: 1,142.28
Iteración 6000/20000 Mejor costo hasta ahora: 1,123.89
Iteración 8000/20000 Mejor costo hasta ahora: 1,098.88
Iteración 10000/20000 Mejor costo hasta ahora: 1,089.19
Iteración 12000/20000 Mejor costo hasta ahora: 1,077.78
Iteración 14000/20000 Mejor costo hasta ahora: 1,074.94
Iteración 16000/20000 Mejor costo hasta ahora: 1,074.94
Iteración 18000/20000 Mejor costo hasta ahora: 1,074.94
Iteración 20000/20000 Mejor costo hasta ahora: 1,074.27

--- Recocido completado ---
Mejor costo de ruta encontrado: 1,074.27
```

**Análisis:** Al permitir una búsqueda exhaustiva, el algoritmo converge a un costo muy cercano al óptimo global (\$1,024.82 en otra ejecución). El costo llega rápidamente a valores cercanos al óptimo, pasando el resto de las iteraciones en el proceso de refinamiento final.

### ***Ejemplo de las rutas de los primeros 3 centros***

```
Centro de Distribución 1 (5 tiendas):
Ruta final: Tienda 38 → Tienda 34 → Tienda 49 → Tienda 30 → Tienda 45
Costo Inicial: $40,258.75
Costo Final: $63.05 (Mejora: $40,195.69)

Centro de Distribución 2 (6 tiendas):
Ruta final: Tienda 1 → Tienda 6 → Tienda 74 → Tienda 81 → Tienda 47 → Tienda 69
Costo Inicial: $20,137.37
Costo Final: $75.82 (Mejora: $20,061.54)

Centro de Distribución 3 (4 tiendas):
Ruta final: Tienda 26 → Tienda 79 → Tienda 88 → Tienda 17
Costo Inicial: $80,247.50
Costo Final: $56.50 (Mejora: $80,191.00)
```

### **Prueba 2 (Búsqueda Restringida)**

- **Parámetros:** *Tinicial* = 500, 100 iteraciones.

- **Mejor Costo Encontrado:** \$202,261.72 (Aún alto debido a asignaciones no corregidas).

Costo de la solución inicial: 302,387.95  
 --- Recocido completado ---  
 Mejor costo de ruta encontrado: 202,261.72

**Análisis:** La reducción drástica de la temperatura y las iteraciones limita el tiempo que el algoritmo tiene para salir de los óptimos locales.

### ***Ejemplo de las rutas de los primeros 3 centros***

Centro de Distribución 1 (7 tiendas):  
 Ruta final: Tienda 30 → Tienda 34 → Tienda 38 → Tienda 45 → Tienda 48 → Tienda 51 → Tienda 49  
 Costo Inicial: \$60,186.74  
 Costo Final: \$40,212.33 (Mejora: \$19,974.41)

Centro de Distribución 2 (7 tiendas):  
 Ruta final: Tienda 1 → Tienda 6 → Tienda 47 → Tienda 81 → Tienda 84 → Tienda 69 → Tienda 82  
 Costo Inicial: \$60,345.39  
 Costo Final: \$40,254.32 (Mejora: \$20,091.07)

Centro de Distribución 3 (5 tiendas):  
 Ruta final: Tienda 62 → Tienda 26 → Tienda 88 → Tienda 17 → Tienda 79  
 Costo Inicial: \$20,133.36  
 Costo Final: \$20,132.80 (Mejora: \$0.56)

El algoritmo se enfoca en perfeccionar el **orden de movimiento** dentro de rutas ya preasignadas a centros cercanos, lo cual es la clave de la optimización.

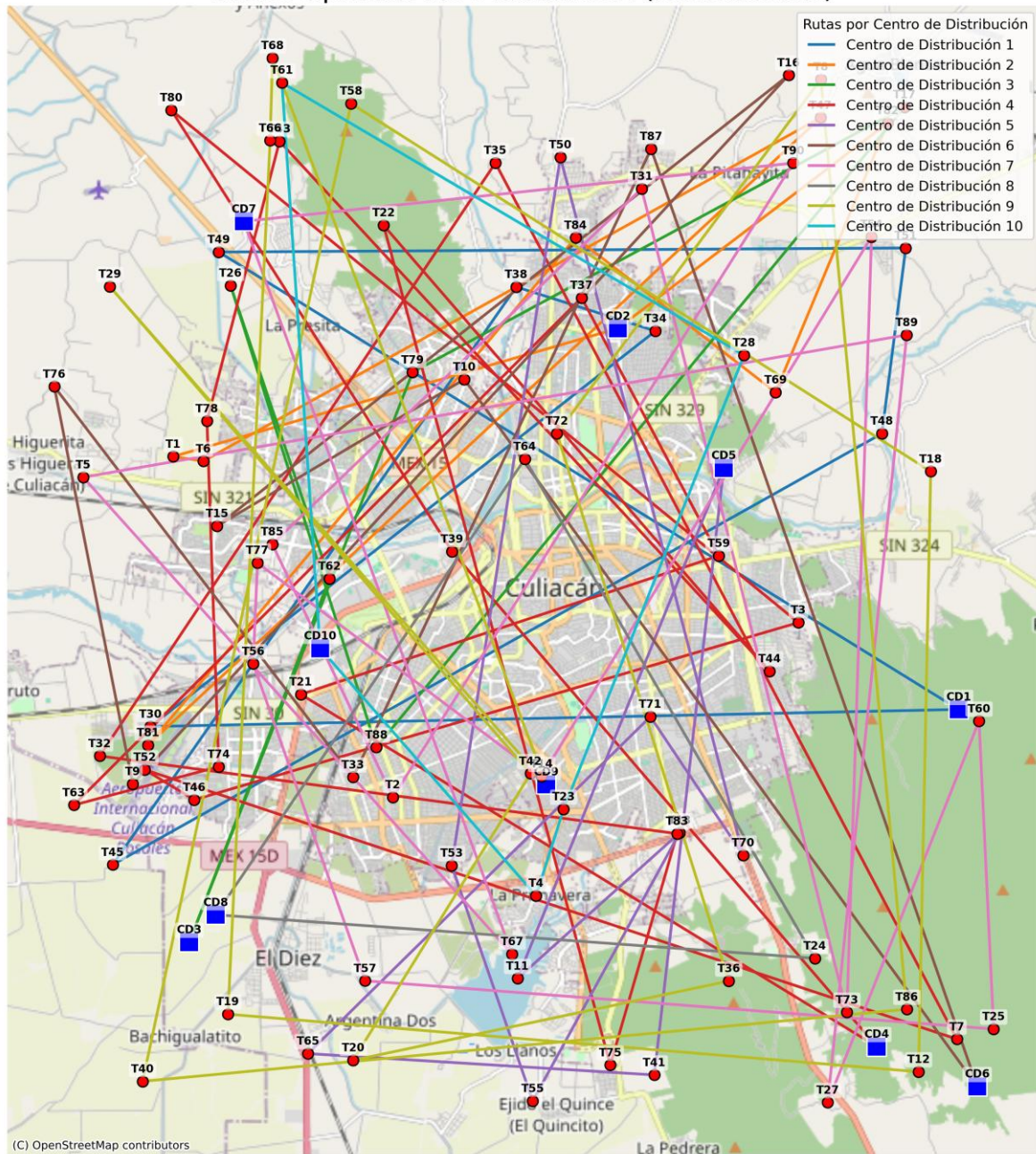
Las pruebas comparativas demuestran que, si bien la cantidad de tiendas asignadas por centro puede ser similar incluso con menos iteraciones, el **tiempo de ejecución es crucial**:

- **Búsqueda Exhaustiva (Prueba 1):** Un tiempo suficiente permitió al algoritmo reducir drásticamente el costo del **Centro de Distribución 1** de \$40,258 a \$63 (con 5 tiendas).
- **Búsqueda Restringida (Prueba 2):** Con menos iteraciones, el costo del **Centro de Distribución 1** solo pasó de \$60,186 a \$40,212 (con 7 tiendas).

La diferencia de costos de más de \$40,000 entre las dos soluciones finales, causada por una pequeña variación en la ruta (solo 2 tiendas más y el orden), resalta que el **orden de visita** afecta el costo de manera extraordinaria. Por lo tanto, es indispensable **dar al algoritmo tiempo suficiente de ejecución** para garantizar ganancias espectaculares.



## Rutas Óptimas de Distribución (Consolidado)



**Evolución del Costo:** El gráfico ilustra la característica clave del algoritmo: los picos en las primeras iteraciones representan la **aceptación de soluciones peores**, lo que permite al algoritmo escapar de óptimos locales, resultando en una mejor solución global al final.

## Conclusiones

El proyecto demostró con éxito la efectividad del **Recocido Simulado** en la resolución de un problema logístico real, al permitir la identificación de rutas eficientes que minimizan el costo total.

1. **Validación del Método:** El enfoque heurístico se justificó y validó, logrando una reducción dramática del costo al corregir las asignaciones no óptimas.
2. **Robustez del Modelo:** La incorporación de una **Penalidad Geográfica** en la función de costo fortaleció el modelo, forzándolo a generar soluciones robustas que respetan la asignación óptima de tiendas con base en la cercanía.
3. **Implementación Técnica:** La implementación modular en **Python**, con visualización geoespacial (**GeoPandas**) y la integración de una interfaz **GTK 4** que utiliza **redirección de `sys.stdout`** para mostrar la consola en tiempo real, facilitó la validación del proceso.

Como trabajo futuro, se podría explorar la integración de una metaheurística híbrida, combinando el Recocido Simulado con la Búsqueda Tabú (Osman, 1993) o evaluando su desempeño frente a algoritmos genéticos para el MDVRP (Centeno Romero & Velásquez, 2016).

## Referencias

- Álvarez Martínez, D., Toro Ocampo, E., & Gallego Rendón, R. (2008). *Algoritmo recocido simulado aplicado al problema de secuenciamiento de tareas en sistemas de producción lineal Flow-Shop* (Vol. XIV). Pereira, Colombia: Scientia Et Technica. Obtenido de <https://www.redalyc.org/articulo.oa?id=84920454030>
- Applegate, D. L., Bixby, R. E., Chvátal, V., & Cook, W. J. (2011). *The Traveling Salesman Problem: A Computational Study*. Princeton University Press.
- Centeno Romero, M., & Velásquez, R. (2016). *Un algoritmo metaheurístico de recocido simulado para eel 3AP-AXIAL* (Vol. 28). Venezuela: Revista Multidisciplinaria del Consejo de Investigación de la Universidad de Oriente. Obtenido de <https://www.redalyc.org/articulo.oa?id=427750771013>
- Kaggle. (s.f.). *Learn Pandas Tutorials*. Obtenido de Kaggle: <https://www.kaggle.com/learn/pandas>
- Osman, I. H. (1993). *Meta-strategy simulated annealing and Tabu search algorithms for the vehicle routine problem* (Vol. 41). Ann Oper Res. doi:10.1007/BF02023004
- Rodríguez-Vásquez, W. C. (2019). *Modelado de un problema de ruteo de vehículos con múltiples depósitos, ventanas de tiempo y flota heterogénea de un servicio de mensajería*. Bogotá, Colombia: Universitaria Agustiniana, Facultad de Ingeniería. Obtenido de [https://www.scielo.cl/scielo.php?script=sci\\_arttext&pid=S0718-07642020000100207&lng=en&nrm=iso&tlng=en](https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-07642020000100207&lng=en&nrm=iso&tlng=en)

## Link – Repositorio

[https://github.com/AnaKMLopez/ITCAKML-Topicos-IA/tree/main/MODULO%202/Recocido Rutas](https://github.com/AnaKMLopez/ITCAKML-Topicos-IA/tree/main/MODULO%202/Recocido%20Rutas)