

# Taller Práctico 1: Procesamiento Paralelo y el Problema del Viajero

Por: Guillermo Andres De Mendoza Corrales  
Universidad Sergio Arboleda, Octubre 2025

---

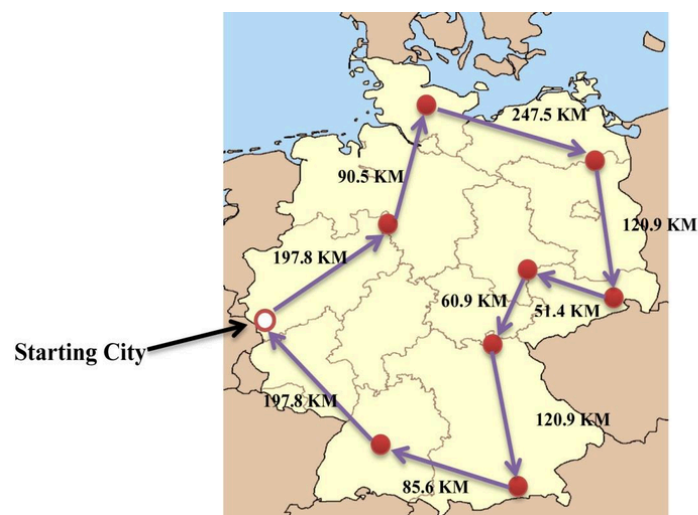
## Objetivo del Laboratorio

El objetivo de este laboratorio es implementar y comparar la eficiencia de un algoritmo **secuencial** y uno **paralelo** (utilizando múltiples procesos) para resolver el Problema del Viajante (Travelling Salesperson Problem, **TSP**) mediante el método de fuerza bruta. El resultado de este laboratorio será un informe que demuestre el *speedup* (aceleración) logrado con el paralelismo.

---

## 1. Fundamentos Teóricos: El Problema del Viajero y la Fuerza Bruta

El **problema del vendedor viajero** (problema del vendedor ambulante, problema del agente viajero o problema del viajante, **PCP**, **TSP** por sus siglas en inglés, Travelling Salesman Problem) responde a la siguiente pregunta: dada una lista de ciudades y las distancias entre cada par de ellas, ¿cuál es la ruta más corta posible que visita cada ciudad exactamente una vez y al finalizar regresa a la ciudad origen? Este es un problema NP-Hard dentro en la optimización combinatoria, muy importante en investigación operativa y en ciencias de la computación.



El problema fue formulado por primera vez en 1930 y es uno de los problemas de optimización más estudiados. Es usado como prueba para muchos métodos de optimización. Aunque el problema es computacionalmente complejo, se conoce gran

cantidad de heurísticas y métodos exactos, así que es posible resolver planteamientos concretos del problema desde cien hasta miles de ciudades.

El TSP tiene diversas aplicaciones aún en su formulación más simple, tales como: la planificación, la logística y la fabricación de circuitos electrónicos. Un poco modificado, aparece como subproblema en muchos campos como la secuenciación de ADN. En esta aplicación, el concepto de “ciudad” representa, por ejemplo: clientes, puntos de soldadura o fragmentos de ADN y el concepto de “distancia” representa el tiempo de viaje o costo, o una medida de similitud entre los fragmentos de ADN. En muchas aplicaciones, restricciones adicionales como el límite de recurso o las ventanas de tiempo hacen el problema considerablemente difícil. El TSP es un caso especial de los Problemas del Comprador Viajante (travelling purchaser problem).

En la teoría de la complejidad computacional, la versión de decisión del TSP (donde, dada una longitud “L”, el objetivo es decidir si el grafo tiene un camino menor o igual que L) pertenece a la clase de los problemas NP-completos. Por tanto, es probable que en el caso peor el tiempo de ejecución para cualquier algoritmo que resuelva el TSP aumente de forma exponencial con respecto al número de ciudades.

## 2. Descripción técnica

En el problema se presentan  $N!$  rutas posibles, aunque se puede simplificar ya que dada una ruta nos da igual el punto de partida y esto reduce el número de rutas a examinar en un factor  $N$  quedando  $(N-1)!$ . Como no importa la dirección en que se desplace el viajante, el número de rutas a examinar se reduce nuevamente en un factor 2. Por lo tanto, hay que considerar  $(N-1)!/2$  rutas posibles.

En la práctica, para un problema del viajante con 5 ciudades hay  $(5-1)!/2=12$  rutas diferentes y no necesitamos un ordenador para encontrar la mejor ruta, pero apenas aumentamos el número de ciudades las posibilidades crece factorialmente:

- Para 10 ciudades hay  $(10-1)!/2=181.440$  rutas diferentes
- Para 30 ciudades hay más de  $4 \cdot 10^{30}$  rutas posibles. Un ordenador que calcule un millón de rutas por segundo necesitaría  $10^{17}$  años para resolverlo. Dicho de otra forma, si se hubiera comenzado a calcular al comienzo de la creación del universo (hace unos 13.400 millones de años) todavía no se habría terminado.

Puede comprobarse que por cada ciudad nueva que incorporamos, el número de rutas se multiplica por el factor  $N$  y crece factorialmente. Por ello el problema pertenece a la clase de problemas NP-completos.

---

### 3. Nomenclatura técnica del problema

3.1 Ciudades: Lista de puntos como coordenadas X , Y. ejemplo: [ [1,1] , [3,5] , [5,9], [0,6] ]

3.2 Ruta: Lista ordenada de ciudades visitadas, ejemplo: [ 2,4,1,3 ] , donde se entiende que iniciamos en la ciudad 2

3.3 Distancia recorrida: valor numerico, el cual evalúa la distancia recorrida en una RUTA, ejemplo: 40.2

---

### 4. Elementos a evaluar con nota academica

- Algoritmos ejecutan correctamente
- Código limpio
- Informe de laboratorio

---

### 5. Algoritmos a desarrollar

Debe crear un algoritmo secuencial y otro paralelo, con el cual encontrar la solución al problema del viajero para de un conjunto de ciudades

Entradas:

- lista de ciudades como una lista de duplas [x,y]

Salida:

- Ruta más corta
- Distancia recorrida en la ruta más corta

---

### 6. Análisis de Resultados y Elaboración del Informe de Laboratorio

El análisis es la parte más importante. Debes documentar los resultados en formato de informe técnico.

Presenta los siguientes puntos en un documento estilo informe de laboratorio:

1. **Título y Objetivos:** (Ya definidos).
2. **Marco Teórico:** Brevemente, explica por qué el TSP es difícil y cómo el paralelismo con fuerza bruta es un caso de estudio ideal.
3. **Metodología:**
  - o **Datos:** Incluye la tabla de ciudades utilizada.
  - o **Configuración del Hardware:** Indica el número de núcleos/procesadores de la máquina utilizada.
  - o **Configuración del Software:** Versiones de Python y librerías
  - o **Explicación de los algoritmos desarrollados.**
4. **Resultados:** Presenta tus hallazgos de los dos algoritmos,

5. **Análisis de Rendimiento:** Compara los resultados obtenidos.
6. **Conclusiones:** Resume tus aprendizajes sobre la aplicación práctica del procesamiento paralelo.