

UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL ROSARIO

**Materia:** Algoritmos Genéticos

**Comisión:** 3EK03

---

## PROBLEMA DEL VIAJANTE (TSP)

---

### Trabajo Práctico N°3

**Ciclo Lectivo:** 2025

**Integrantes del grupo:**

<b>Nombre y Apellido</b>	<b>Correo electrónico</b>	<b>Legajo</b>
Juan Cruz Mondino	juancm.2000@hotmail.com	51922
Alexis Mateo	alexisjoelmateo@gmail.com	51191
Gustavo Giampietro	gustgiam2001@gmail.com	50671

**Profesores:**

Daniela Díaz  
Víctor Lombardo

# Índice

<b>1 Enunciado del Problema del Viajante</b>	2
1.1 Ejercicios . . . . .	2
1.2 Recomendaciones para el algoritmo genético . . . . .	2
<b>2 Búsqueda Exhaustiva</b>	3
2.1 Conclusión del ejercicio . . . . .	3
<b>3 Algoritmo Genético</b>	3
3.1 Función Objetivo . . . . .	3
3.2 Codificación y Operadores . . . . .	3
3.3 Datos de ingreso . . . . .	4
3.4 Resultado aplicando selección con ruleta sin elitismo . . . . .	5
3.5 Resultado aplicando selección con ruleta con elitismo . . . . .	6
3.6 Resultado aplicando selección con torneo sin elitismo . . . . .	7
3.7 Resultado aplicando selección con torneo con elitismo . . . . .	8
3.8 Resultado aplicando selección con torneo con elitismo y con 200000 corridas . . . . .	9
<b>4 Algoritmo Heurístico</b>	9
<b>5 Aportes Prácticos del TSP</b>	10
5.1 Logística de última milla y distribución urbana . . . . .	10
5.2 Ruteo de servicios técnicos y lecturas de medidores . . . . .	11
5.3 Planificación de recorridos turísticos . . . . .	11
5.4 Manufactura y maquinado (CNC / corte láser / plotters) . . . . .	11
5.5 Inspección con drones y robots móviles . . . . .	11
5.6 Diseño y prueba de circuitos impresos (PCB) y <i>chip testing</i> . . . . .	11
5.7 Planificación de recolección de residuos y barrido . . . . .	11
<b>6 Conclusiones</b>	12
<b>7 Código Fuente</b>	12

# 1. Enunciado del Problema del Viajante

El *Problema del Viajante* (TSP) busca una ruta que, comenzando y terminando en una misma ciudad, visite exactamente una vez cada una de las  $N$  ciudades y minimice la distancia total recorrida. Las distancias están dadas por una matriz  $D$  de tamaño  $N \times N$ , donde  $d[x, y]$  representa la distancia entre la ciudad  $x$  y la ciudad  $y$ .

## 1.1. Ejercicios

### Ejercicios a resolver:

1. Hallar la ruta de distancia mínima que logre unir todas las capitales de provincias de la República Argentina mediante un método **exhaustivo**. ¿Puede resolverse? Justificar teóricamente.
2. Desarrollar un programa con menú que permita:
  - 2.a) Ingresar una provincia inicial y hallar la ruta con **heurística del vecino más cercano** (ir a la ciudad no visitada más próxima), retornando a la ciudad de partida. Mostrar mapa, ciudad de partida, recorrido completo y longitud del trayecto.
  - 2.b) Encontrar el **recorrido mínimo** global aplicando la misma heurística. Mostrar recorrido y longitud.
  - 2.c) Hallar la ruta mínima utilizando un **algoritmo genético**.
3. **Comparar** resultados entre la heurística y el algoritmo genético, anexando la conclusión.
4. Agregar el apartado «**Aportes Prácticos del TSP**» con **al menos dos** aplicaciones actuales y su explicación.

## 1.2. Recomendaciones para el algoritmo genético

- $N = 50$  cromosomas por población.
- $M = 200$  ciclos/iteraciones.
- Codificación: permutaciones de 23 números (1..23), cada gen es una ciudad.
- Probabilidades de cruce y mutación: a criterio del grupo.
- **Usar crossover cíclico**.

## 2. Búsqueda Exhaustiva

La búsqueda exhaustiva para el TSP consiste en evaluar *todas* las rutas Hamiltonianas que parten de una ciudad, visitan cada ciudad exactamente una vez y regresan al origen, quedándose con la de menor longitud total.

### 2.1. Conclusión del ejercicio

**No es factible** resolver el TSP de las 24 capitales por búsqueda exhaustiva en tiempo razonable con equipamiento estándar. Por lo tanto, la cátedra propone para este trabajo el uso de métodos aproximados (heurísticas y algoritmos genéticos) para obtener soluciones de buena calidad en tiempos prácticos.

## 3. Algoritmo Genético

### 3.1. Función Objetivo

Definimos la **función** objetivo como proporcional a la distancia:

$$f(x) = \frac{1}{\text{distancia}}$$

### 3.2. Codificación y Operadores

Las probabilidades de cruce y **mutación** son:

- Probabilidad de **mutación**: 5 %
- Probabilidad de **cruce**: 75 %

### 3.3. Datos de ingreso

Las ciudades tienen asociado un **número**:

Número	Ciudad
0	Cdad. de Bs. As.
1	Córdoba
2	Corrientes
3	Formosa
4	La Plata
5	La Rioja
6	Mendoza
7	Neuquén
8	Paraná
9	Posadas
10	Rawson
11	Resistencia
12	Río Gallegos
13	S.F.d.V.d. Catamarca
14	S.M. de Tucumán
15	S.S. de Jujuy
16	Salta
17	San Juan
18	San Luis
19	Santa Fe
20	Santa Rosa
21	Sgo. del Estero
22	Ushuaia
23	Viedma

### 3.4. Resultado aplicando selección con ruleta sin elitismo

Número	Ciudad	Número	Ciudad
0	San Luis	13	Viedma
1	La Rioja	14	San Miguel de Tucumán
2	San Juan	15	Resistencia
3	Mendoza	16	Corrientes
4	Posadas	17	Formosa
5	Salta	18	Santa Fe
6	San Salvador de Jujuy	19	Santa Rosa
7	San Fernando del Valle de Catamarca	20	La Plata
8	Santiago del Estero	21	CABA
9	Córdoba	22	Paraná
10	Río Gallegos	23	Neuquén
11	Ushuaia	24	San Luis
12	Rawson	—	—

Esta **solución** devuelve un total de 14997 km en un total de 0.672346 s.

En la siguiente **gráfica** se muestra la **evolución** de la función objetivo:

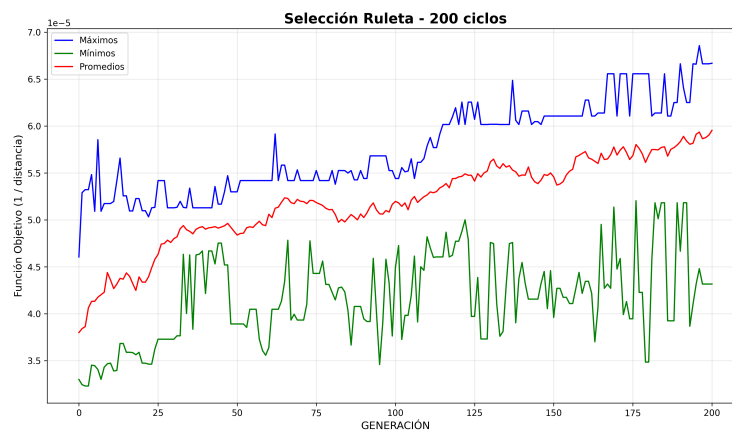


Figura 1: Ruleta.

### 3.5. Resultado aplicando selección con ruleta con elitismo

Nº	Ciudad	Nº	Ciudad
0	La Plata	13	San Miguel de Tucumán
1	Paraná	14	San Juan
2	Santa Fe	15	Santa Rosa
3	Ushuaia	16	Viedma
4	Río Gallegos	17	Rawson
5	Neuquén	18	Mendoza
6	San Luis	19	La Rioja
7	Córdoba	20	San Fernando del Valle de Catamarca
8	Corrientes	21	Formosa
9	Resistencia	22	Posadas
10	Santiago del Estero	23	CABA
11	San Salvador de Jujuy	24	La Plata
12	Salta	—	—

Esta **solución** devuelve un total de 13536 km en un total de 0.636614 s.

En la siguiente **gráfica** se muestra la **evolución** de la función objetivo:

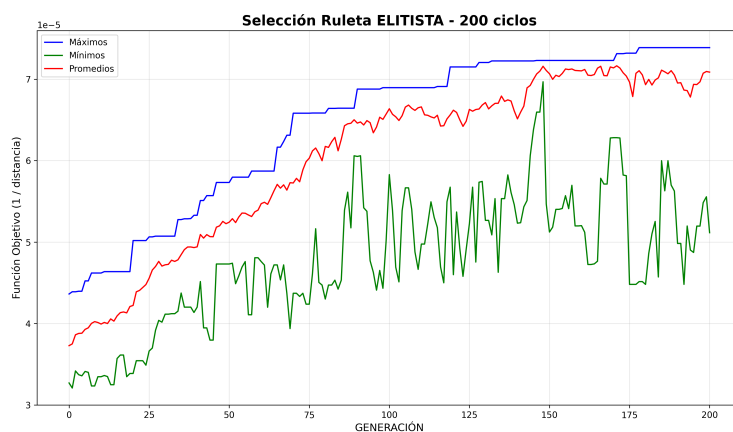


Figura 2: Ruleta elitista.

### 3.6. Resultado aplicando selección con torneo sin elitismo

Nº	Ciudad	Nº	Ciudad
0	Paraná	13	Rawson
1	CABA	14	Viedma
2	La Plata	15	Posadas
3	Santa Rosa	16	Corrientes
4	Neuquén	17	Resistencia
5	San Luis	18	Formosa
6	La Rioja	19	Córdoba
7	Salta	20	San Juan
8	San Salvador de Jujuy	21	San Fernando del Valle de Catamarca
9	Santiago del Estero	22	San Miguel de Tucumán
10	Mendoza	23	Santa Fe
11	Río Gallegos	24	Paraná
12	Ushuaia		

Esta **solución** devuelve un total de 11722 km en un total de 0.810403 s.

En la siguiente **gráfica** se muestra la **evolución** de la función objetivo:

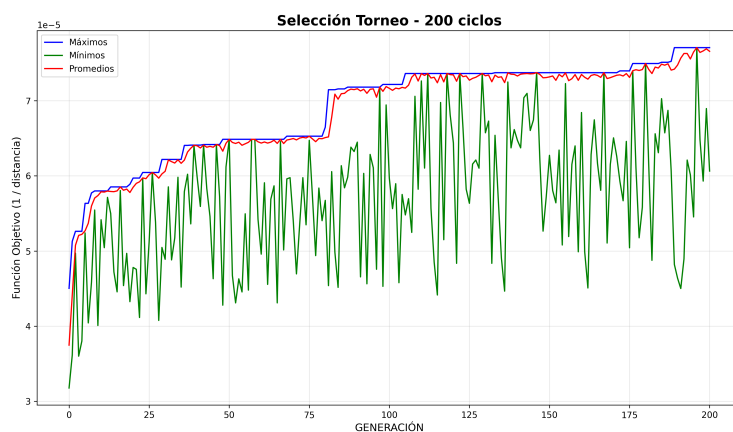


Figura 3: Torneo.



### 3.7. Resultado aplicando selección con torneo con elitismo

Nº	Ciudad	Nº	Ciudad
0	Formosa	13	Mendoza
1	Posadas	14	Santiago del Estero
2	Corrientes	15	San Luis
3	Resistencia	16	Neuquén
4	Paraná	17	Río Gallegos
5	Santa Fe	18	Ushuaia
6	Córdoba	19	Rawson
7	San Miguel de Tucumán	20	Santa Rosa
8	San Salvador de Jujuy	21	Viedma
9	Salta	22	CABA
10	La Rioja	23	La Plata
11	San Fernando del Valle de Catamarca	24	Formosa
12	San Juan		

Esta **solución** devuelve un total de 11665 km en un total de 0.828629 s.

En la siguiente **gráfica** se muestra la **evolución** de la función objetivo:

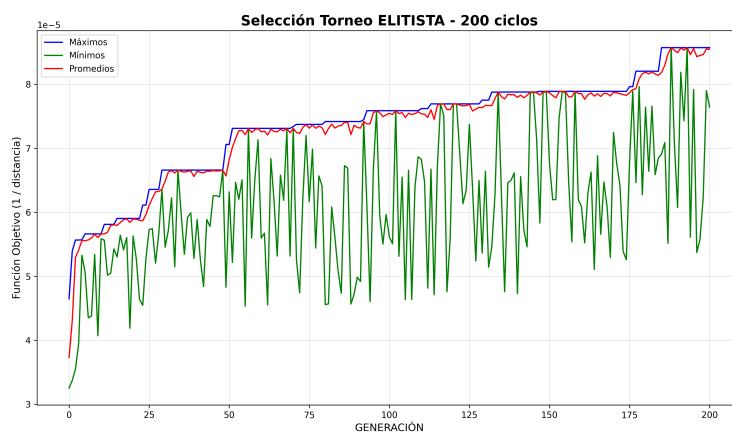


Figura 4: Torneo elitista.

### 3.8. Resultado aplicando selección con torneo con elitismo y con 200000 corridas

Nº	Ciudad	Nº	Ciudad
0	Resistencia	13	La Rioja
1	Corrientes	14	San Juan
2	Formosa	15	Mendoza
3	Posadas	16	San Luis
4	La Plata	17	Santa Rosa
5	CABA	18	Neuquén
6	Santa Fe	19	Río Gallegos
7	Córdoba	20	Ushuaia
8	Santiago del Estero	21	Rawson
9	San Salvador de Jujuy	22	Viedma
10	Salta	23	Paraná
11	San Miguel de Tucumán	24	Resistencia
12	San Fernando del Valle de Catamarca	–	–

Esta **solución** devuelve un total de 9942 km en un total de 258 s.

## 4. Algoritmo Heurístico

La heurística aplicada parte de una ciudad inicial seleccionada por el usuario y, en cada paso, elige la **capital no visitada más cercana** según la matriz de distancias. Este procedimiento se repite hasta recorrer todas las ciudades exactamente una vez y, finalmente, se retorna a la ciudad de partida para cerrar el circuito.

Operativamente, el algoritmo mantiene el conjunto de ciudades no visitadas y, tras cada selección, **elimina** la ciudad recién incorporada del conjunto de candidatas. La decisión es puramente local (codiciosa): se prioriza el menor costo inmediato sin analizar globalmente el resto del recorrido. Por esta razón, la heurística no garantiza la óptima global, pero ofrece **buenos tiempos de cómputo** y una implementación simple.

El resultado **depende de la ciudad inicial**. Para evaluar la sensibilidad a la elección del origen, se ejecuta la heurística iniciando desde cada capital posible y se registra, en cada caso, el recorrido completo y la distancia total. En la tabla que sigue se muestran los **recorridos y costos** obtenidos para cada inicio.

La correspondencia entre **índices numéricos y nombres de ciudades** utilizada en los recorridos figura más arriba en este documento. Salvo indicación contraria, las distancias provienen de la matriz simétrica de kilómetros entre capitales. En caso de empates en la distancia mínima, se aplica una regla determinística (por ejemplo, el índice menor) para asegurar **reproducibilidad** de los resultados.

Inicio	Recorrido	KM
7	[7, 20, 18, 6, 17, 5, 13, 21, 14, 16, 15, 11, 2, 3, 9, 8, 19, 1, 0, 4, 23, 10, 12, 22, 7]	9335
10	[10, 23, 20, 18, 6, 17, 5, 13, 21, 14, 16, 15, 11, 2, 3, 9, 8, 19, 1, 0, 4, 7, 12, 22, 10]	9579
12	[12, 22, 10, 23, 20, 18, 6, 17, 5, 13, 21, 14, 16, 15, 11, 2, 3, 9, 8, 19, 1, 0, 4, 7, 12]	9579
22	[22, 12, 10, 23, 20, 18, 6, 17, 5, 13, 21, 14, 16, 15, 11, 2, 3, 9, 8, 19, 1, 0, 4, 7, 22]	9606
1	[1, 18, 6, 17, 5, 13, 21, 14, 16, 15, 11, 2, 3, 9, 8, 19, 0, 4, 20, 7, 23, 10, 12, 22, 1]	9781
4	[4, 0, 8, 19, 1, 18, 6, 17, 5, 13, 21, 14, 16, 15, 11, 2, 3, 9, 20, 7, 23, 10, 12, 22, 4]	9858
18	[18, 6, 17, 5, 13, 21, 14, 16, 15, 11, 2, 3, 9, 8, 19, 1, 20, 7, 23, 10, 12, 22, 4, 0, 18]	9881
8	[8, 19, 1, 18, 6, 17, 5, 13, 21, 14, 16, 15, 11, 2, 3, 9, 0, 4, 20, 7, 23, 10, 12, 22, 8]	9911
0	[0, 4, 8, 19, 1, 18, 6, 17, 5, 13, 21, 14, 16, 15, 11, 2, 3, 9, 20, 7, 23, 10, 12, 22, 0]	9933
19	[19, 8, 1, 18, 6, 17, 5, 13, 21, 14, 16, 15, 11, 2, 3, 9, 0, 4, 20, 7, 23, 10, 12, 22, 19]	9935
6	[6, 17, 5, 13, 21, 14, 16, 15, 11, 2, 3, 9, 8, 19, 1, 18, 20, 7, 23, 10, 12, 22, 4, 0, 6]	10010
23	[23, 10, 7, 20, 18, 6, 17, 5, 13, 21, 14, 16, 15, 11, 2, 3, 9, 8, 19, 1, 0, 4, 12, 22, 23]	10021
20	[20, 18, 6, 17, 5, 13, 21, 14, 16, 15, 11, 2, 3, 9, 8, 19, 1, 0, 4, 23, 10, 7, 12, 22, 20]	10173
15	[15, 16, 14, 21, 13, 5, 17, 6, 18, 1, 19, 8, 0, 4, 20, 7, 23, 10, 12, 22, 11, 2, 3, 9, 15]	10225
5	[5, 13, 21, 14, 16, 15, 11, 2, 3, 9, 8, 19, 1, 18, 6, 17, 20, 7, 23, 10, 12, 22, 4, 0, 5]	10237
14	[14, 21, 13, 5, 17, 6, 18, 1, 19, 8, 0, 4, 20, 7, 23, 10, 12, 22, 11, 2, 3, 9, 16, 15, 14]	10275
16	[16, 15, 14, 21, 13, 5, 17, 6, 18, 1, 19, 8, 0, 4, 20, 7, 23, 10, 12, 22, 11, 2, 3, 9, 16]	10275
21	[21, 14, 13, 5, 17, 6, 18, 1, 19, 8, 0, 4, 20, 7, 23, 10, 12, 22, 11, 2, 3, 9, 16, 15, 21]	10419
13	[13, 5, 17, 6, 18, 1, 19, 8, 0, 4, 20, 7, 23, 10, 12, 22, 21, 14, 16, 15, 11, 2, 3, 9, 13]	10647
17	[17, 6, 18, 1, 19, 8, 0, 4, 20, 7, 23, 10, 12, 22, 5, 13, 21, 14, 16, 15, 11, 2, 3, 9, 17]	10661
9	[9, 3, 2, 11, 19, 8, 1, 18, 6, 17, 5, 13, 21, 14, 16, 15, 0, 4, 20, 7, 23, 10, 12, 22, 9]	10767
11	[11, 2, 3, 9, 8, 19, 1, 18, 6, 17, 5, 13, 21, 14, 16, 15, 0, 4, 20, 7, 23, 10, 12, 22, 11]	10835
2	[2, 11, 3, 9, 8, 19, 1, 18, 6, 17, 5, 13, 21, 14, 16, 15, 0, 4, 20, 7, 23, 10, 12, 22, 2]	10840
3	[3, 2, 11, 9, 8, 19, 1, 18, 6, 17, 5, 13, 21, 14, 16, 15, 0, 4, 20, 7, 23, 10, 12, 22, 3]	11031

Tabla 1: Recorridos ordenados de menor a mayor por cantidad de km

Los resultados muestran que el **mejor recorrido** obtenido por la heurística corresponde al circuito que **inicia en la ciudad 7**, con una distancia total de **9335 km**. Este valor es el mínimo entre todos los inicios evaluados, confirmando que la calidad de la solución **depende de la ciudad de partida**.

## 5. Aportes Prácticos del TSP

El *Problema del Viajante* (TSP) aparece como subproblema o como modelo central en numerosos contextos reales. A continuación se describen aplicaciones típicas, indicando cómo se mapean al TSP y qué decisiones optimiza.

### 5.1. Logística de última milla y distribución urbana

**Descripción:** Determinar el orden de visita de puntos de entrega/retiro (domicilios, comercios) partiendo de un depósito y retornando al mismo.

**Modelo TSP:** Cada punto de entrega es una “ciudad” y la distancia/coste es el tiempo de viaje o la distancia vial.

**Objetivo:** Minimizar kilómetros/tiempo total y costos operativos, mejorando la puntualidad y reduciendo combustible.

## **5.2. Ruteo de servicios técnicos y lecturas de medidores**

**Descripción:** Técnicos que deben visitar un conjunto de clientes (reparaciones, mantenimiento preventivo) o postes/medidores para lectura.

**Modelo TSP:** Los sitios a visitar son nodos; las aristas ponderan desplazamientos entre sitios.

**Objetivo:** Minimizar traslados para aumentar cantidad de servicios diarios y disminuir costos.

## **5.3. Planificación de recorridos turísticos**

**Descripción:** Definir un circuito que recorra atractivos turísticos (city tours) partiendo y terminando en un punto.

**Modelo TSP:** Atracciones como nodos; tiempos de traslado (o combinados con penalizaciones por congestión) en aristas.

**Objetivo:** Minimizar tiempo de desplazamiento para maximizar tiempo efectivo en atracciones.

## **5.4. Manufactura y maquinado (CNC / corte láser / plotters)**

**Descripción:** Orden óptimo de operaciones o cortes para minimizar movimientos en vacío del cabezal/herramienta.

**Modelo TSP:** Cada operación (agujero, contorno) es un nodo; el “viaje” entre operaciones es el reposicionamiento.

**Objetivo:** Reducir tiempos muertos y desgaste, elevando productividad.

## **5.5. Inspección con drones y robots móviles**

**Descripción:** Recorrido que visita múltiples puntos de inspección (torres, techos, paneles solares) y retorna a base.

**Modelo TSP:** Puntos de inspección como nodos; costo por energía/tiempo de vuelo entre nodos.

**Objetivo:** Minimizar consumo energético y respetar ventanas de batería, aumentando cobertura por misión.

## **5.6. Diseño y prueba de circuitos impresos (PCB) y *chip testing***

**Descripción:** Secuencia de *test points* o contactos para sondas, y recorrido de taladros en PCB.

**Modelo TSP:** Puntos de prueba/taladros como nodos; tiempos de movimiento de la sonda/husillo como aristas.

**Objetivo:** Disminuir tiempo de ciclo y aumentar throughput de fabricación/pruebas.

## **5.7. Planificación de recolección de residuos y barrido**

**Descripción:** Circuitos que visitan bocas de recolección o segmentos priorizados de calles.

**Modelo TSP:** Puntos representativos o inicios de tramos como nodos; desplazamientos como aristas.

**Objetivo:** Minimizar el recorrido total manteniendo la cobertura y frecuencia de servicio.

**Notas:** En muchos casos reales, el TSP es un bloque dentro de problemas más amplios (p.ej., VRP con múltiples vehículos, ventanas de tiempo o capacidades). Aun así, resolver (o aproximar) bien el TSP subyacente aporta mejoras directas en distancia/tiempo, productividad y costos.

## 6. Conclusiones

Los resultados obtenidos permiten establecer una comparación cuantitativa y cualitativa entre la búsqueda heurística determinista y los algoritmos genéticos (AG) aplicados al problema del viajante o ruteo entre capitales provinciales.

Desde un punto de vista computacional, la búsqueda heurística presenta una convergencia inmediata hacia una solución con un costo temporal mínimo, mientras que el algoritmo genético, si bien logra aproximarse a soluciones comparables, requiere varios órdenes de magnitud más iteraciones (en nuestro caso, recién cuando se probó con 200000 generaciones se obtuvieron, a veces, resultados similares) y, por ende, un costo computacional considerablemente mayor.

En nuestro caso, el algoritmo genético está diseñado para maximizar una función objetivo definida como 1 dividido la distancia total del recorrido. De esta manera, cuanto menor es la distancia recorrida, mayor es el valor de la función objetivo, convirtiendo un problema de minimización de distancia en uno de maximización de desempeño.

Por otro lado, la búsqueda heurística utilizada se basa en una lógica de tipo “vecino más cercano”, donde en cada paso se elige la siguiente ciudad más próxima. Este método sigue un proceso determinista y rápido, ya que analiza las opciones localmente y construye una solución aceptable en muy poco tiempo. Sin embargo, su principal limitación es que no garantiza encontrar la mejor solución global, sino una que puede ser simplemente buena o cercana al óptimo.

En contraste, el algoritmo genético combina varios procesos evolutivos (como selección, cruce y mutación) sobre una población de posibles soluciones que se evalúan y mejoran durante un gran número de generaciones. Este enfoque implica un esfuerzo computacional mucho mayor, porque en cada iteración se analizan muchos individuos y se generan nuevas combinaciones.

En síntesis, la búsqueda heurística logró soluciones de menor distancia total en tiempos de milisegundos, mientras que el algoritmo genético, aun tras 200000 generaciones, apenas consiguió aproximaciones comparables, pero con tiempos de ejecución varios órdenes de magnitud mayores.

Desde el punto de vista teórico, esto confirma que los AG son métodos de optimización global basados en exploración estocástica, no diseñados para velocidad sino para evitar atraparse en óptimos locales. Por tanto, su uso se justifica cuando la función de costo es altamente irregular o cuando se requiere explorar un espacio de soluciones amplio, mientras que las heurísticas deterministas son preferibles en contextos donde la estructura del problema permite decisiones locales eficientes y reproducibles.

## 7. Código Fuente

<https://drive.google.com/drive/folders/18ibY6AicsOBgQDaoscLlPqdqnq8yiavY?usp=sharing>