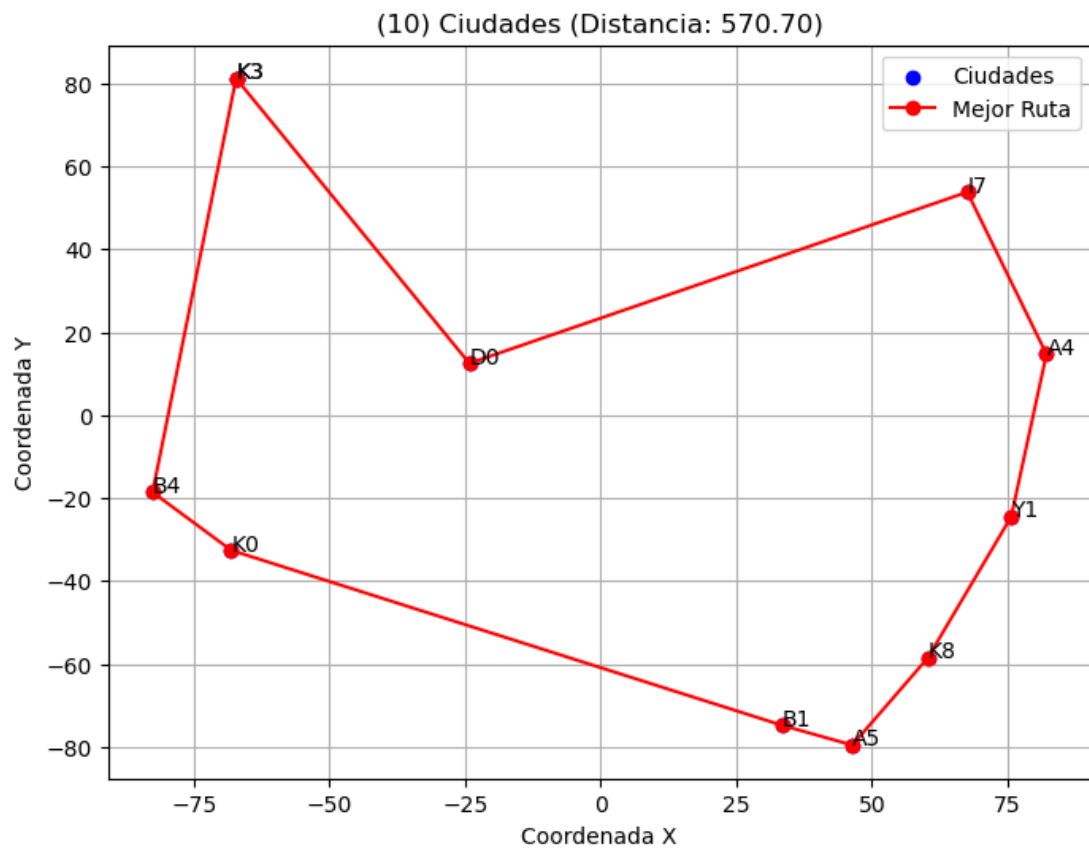
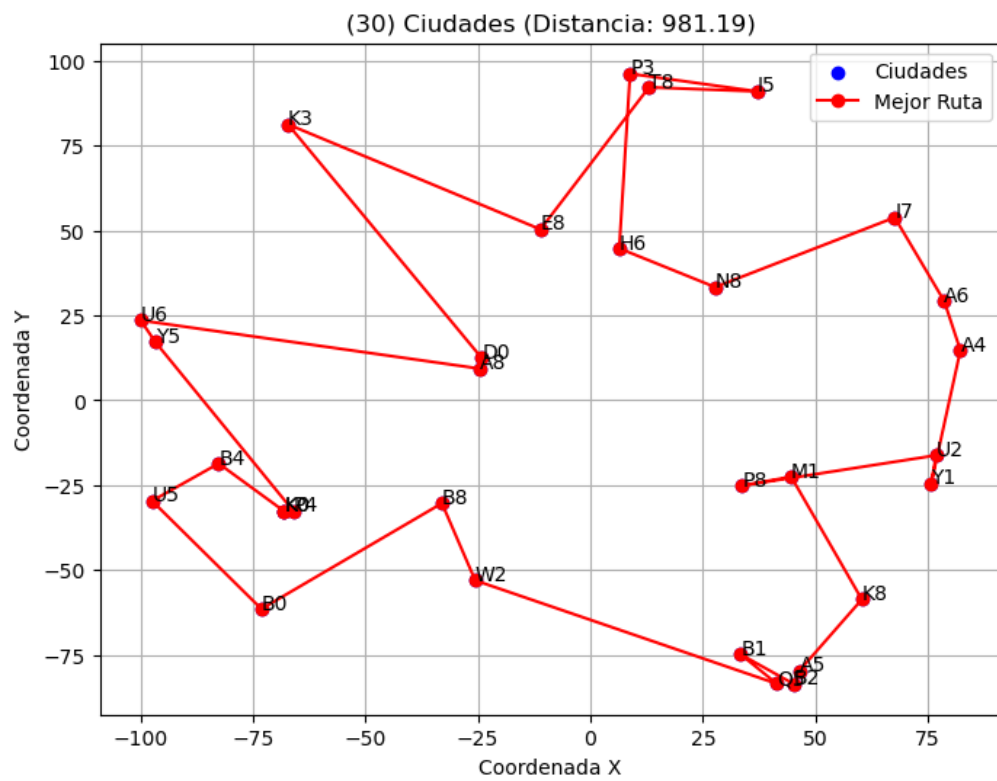
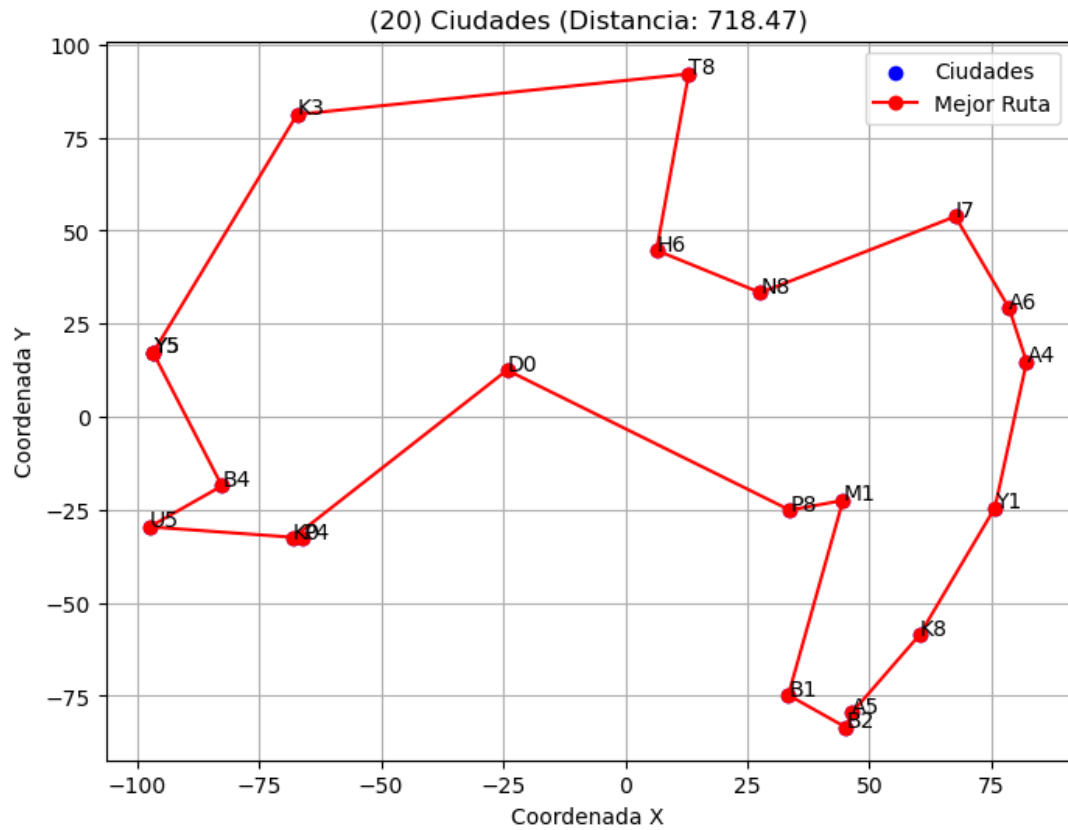
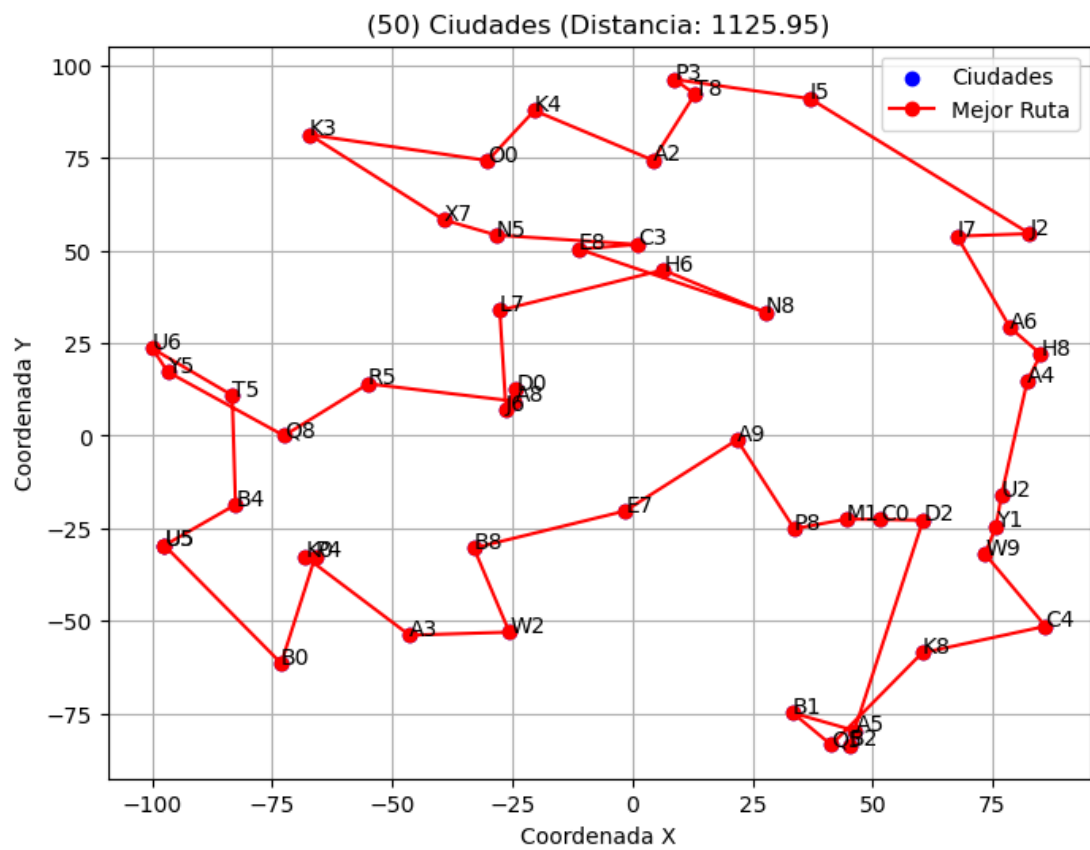
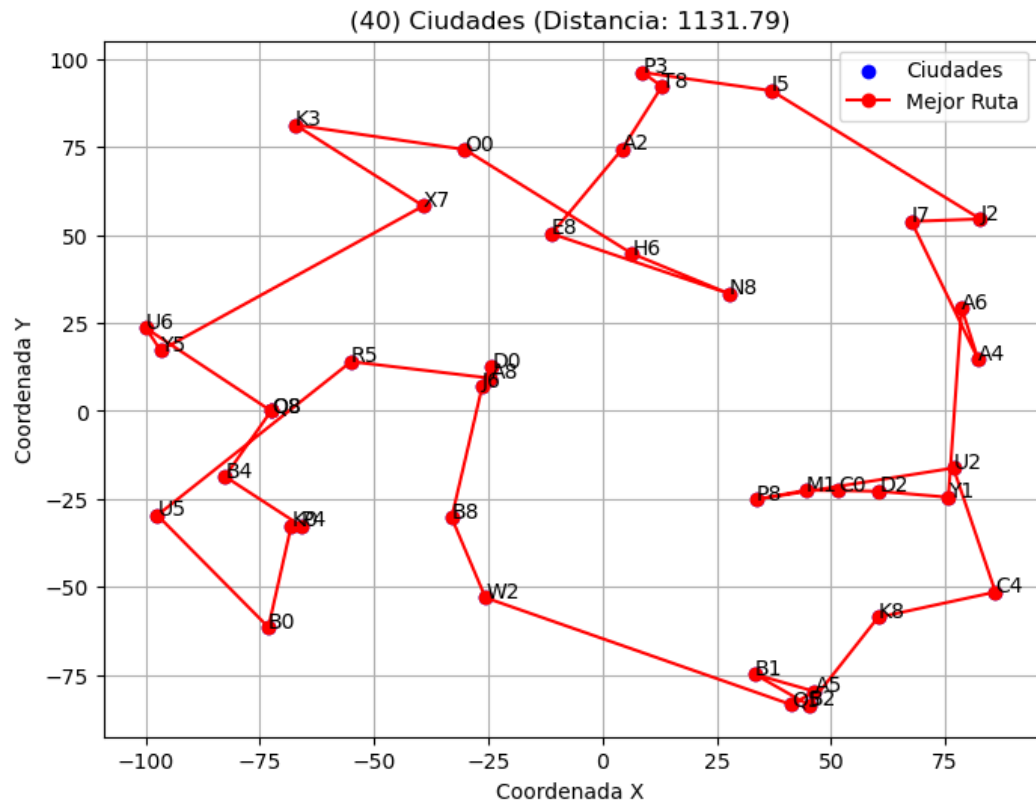


### A. Analizar el código propuesto







Como se puede observar en las corridas anteriores. El número de ciudades es el factor diferencial al momento de tomar tiempos ya que aumenta el procesamiento de la información. Como el tiempo límite sigue establecido a 30 segundos, todos los ejemplos tienen el mismo límite de tiempo.

Al no usarse ninguna heurística, es probable que el algoritmo base sea aleatorio, Greedy sin mejora, o alguna forma de recorrido sin optimización. Esto reduce notablemente la calidad de la solución.

Al escalar el algoritmo del TSP a 10, 20, 30, 40 y 50 ciudades, el impacto sobre los tiempos de ejecución, la calidad de la solución y la viabilidad del enfoque cambia drásticamente, especialmente si no se usan heurísticas.

Usando la heurística de 'vecino cercano' se puede observar una mejora en los tiempos de ejecución. Se sigue sin encontrar la mejor ruta en con las heurísticas aplicadas. Sin embargo, en algunos casos la distancia total recorrida con el resultado de la heurística aplicada es más grande que en la corrida inicial. Por otra parte, los tiempos de ejecución con las primeras ciudades mejora bastante con la heurística aplicada.

## B. Analizar parámetro tee

```
GLPSOL: GLPK LP/MIP Solver, v4.65
Parameter(s) specified in the command line:
--mipgap 0.2 --tmlim 40 --write C:\Users\ccarr\AppData\Local\Temp\tmpaafdk_ut.glpk.raw
--wglp C:\Users\ccarr\AppData\Local\Temp\tmp2x21hxy1.glpk.glp --cpxlp C:\Users\ccarr\AppData\Local\Temp\tmpwyl_wx5z.pyomo
o.lp
Reading problem data from 'C:\Users\ccarr\AppData\Local\Temp\tmpwyl_wx5z.pyomo.lp'...
C:\Users\ccarr\AppData\Local\Temp\tmpwyl_wx5z.pyomo.lp:58473: warning: lower bound of variable 'x9804' redefined
C:\Users\ccarr\AppData\Local\Temp\tmpwyl_wx5z.pyomo.lp:58473: warning: upper bound of variable 'x9804' redefined
4973 rows, 4970 columns, 33673 non-zeros
4970 integer variables, 4900 of which are binary
63373 lines were read
Writing problem data to 'C:\Users\ccarr\AppData\Local\Temp\tmp2x21hxy1.glpk.glp'...
53422 lines were written
GLPK Integer Optimizer, v4.65
4973 rows, 4970 columns, 33673 non-zeros
4970 integer variables, 4900 of which are binary
Preprocessing...
4903 rows, 4900 columns, 33603 non-zeros
4900 integer variables, 4830 of which are binary
Scaling...
A: min|aij| = 6.403e-01 max|aij| = 2.343e+02 ratio = 3.659e+02
GM: min|aij| = 2.680e-01 max|aij| = 3.732e+00 ratio = 1.393e+01
EQ: min|aij| = 7.181e-02 max|aij| = 1.000e+00 ratio = 1.393e+01
2N: min|aij| = 5.538e-02 max|aij| = 1.232e+00 ratio = 2.226e+01
Constructing initial basis...
Size of triangular part is 4902
Solving LP relaxation...
GLPK Simplex Optimizer, v4.65
4903 rows, 4900 columns, 33603 non-zeros
0: obj = 1.143600479e+04 inf = 1.116e+04 (140)
321: obj = 2.129370801e+03 inf = 1.047e-13 (0) 2
* 564: obj = 9.640090924e+02 inf = 2.325e-13 (0) 2
OPTIMAL LP SOLUTION FOUND
```

```

Integer optimization begins...
Long-step dual simplex will be used
+ 564: mip =      not found yet >=          -inf          (1; 0)
+ 3223: mip =      not found yet >=    9.644463219e+02    (191; 1)
+ 5548: >>>>  1.927968228e+03 >=    9.800456415e+02    49.2% (378; 1)
+ 8189: mip =    1.927968228e+03 >=    9.850425983e+02    48.9% (524; 28)
+ 10398: >>>>  1.897781670e+03 >=    9.850425983e+02    48.1% (672; 28)
+ 12598: mip =    1.897781670e+03 >=    9.929721580e+02    47.7% (832; 51)
+ 13998: >>>>  1.635554840e+03 >=    9.929721580e+02    39.3% (917; 51)
+ 16667: mip =    1.635554840e+03 >=    9.941769472e+02    39.2% (939; 339)
+ 19554: mip =    1.635554840e+03 >=    9.952943095e+02    39.1% (1089; 341)
+ 21456: >>>>  1.461902029e+03 >=    9.952943095e+02    31.9% (1220; 342)
+ 21898: mip =    1.461902029e+03 >=    9.953896908e+02    31.9% (914; 1017)
TIME LIMIT EXCEEDED; SEARCH TERMINATED
Time used:   40.2 secs
Memory used: 15.0 Mb (15730480 bytes)
Writing MIP solution to 'C:\Users\ccarr\AppData\Local\Temp\tmpaafdk_ut.glpk.raw'...
9952 lines were written
Tiempo de ejecución: 00:40
Distancia mínima entre nodos: 0.6403124237432862
Distancia máxima entre nodos: 234.3041826344549
Distancia promedio entre nodos: 100.75829712883072
Distancia Total mínima posible: 887.2378335850225
Distancia Total máxima posible: 2129.370800604054
Heurísticas aplicadas: ['limitar_funcion_objetivo']
No se encontró una solución óptima, la siguiente es la mejor solución encontrada:
Distancia total recorrida: 1461.9020286291757

```

El parámetro tee ayuda a monitorear el progreso de optimización. También se puede ver el tiempo límite antes de encontrar una solución óptima.

interpretación:

- Se leyeron 4973 restricciones y 4970 variables
- Algunas advertencias por redefinición de límites de variables.
- Se encontró una solución óptima al LP relajado (964.009).
- El tiempo límite fue alcanzado antes de encontrar una solución óptima.

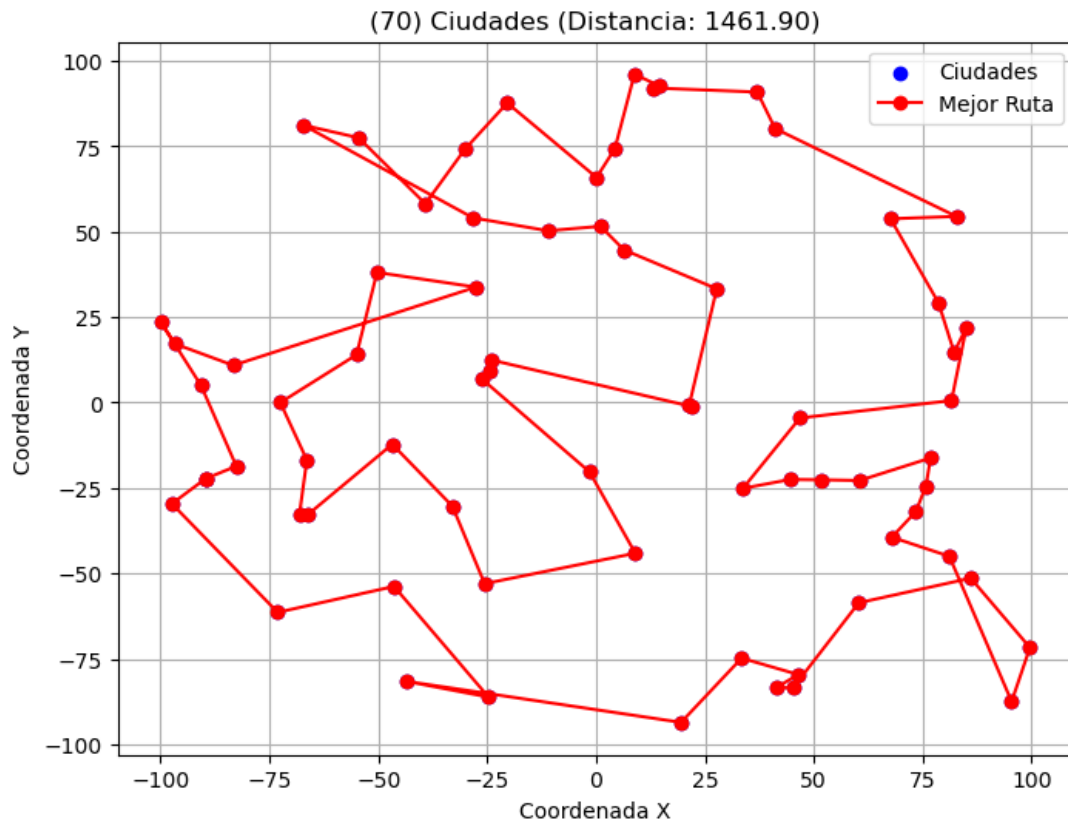
Gap al final:

- El mejor lower bound fue 995.39
- La mejor solución entera (mip) fue 1461.90
- El gap era de 31.9% al finalizar, por eso no es una solución óptima.
- Distancia total recorrida (solución actual): 1461.90
- Distancia mínima posible (teórica o relajada): 887.24
- Distancia máxima posible: 2129.37
- Heurística usada: 'limitar\_funcion\_objetivo': probablemente un recorte para guiar la búsqueda del solver más rápidamente hacia soluciones prometedoras.

Conclusión: El parámetro tee te permitió ver estos logs completos del solver.

El resultado final: No se encontró una solución óptima, pero sí una factible bastante buena. La búsqueda fue interrumpida por tiempo límite.

### C. Aplicar heurística de límites a la función objetivo



#### Con heurística:

Tiempo de ejecución: 00:40

Distancia mínima entre nodos: 0.6403124237432862

Distancia máxima entre nodos: 234.3041826344549

Distancia promedio entre nodos: 100.75829712883072

Distancia Total mínima posible: 887.2378335850225

Distancia Total máxima posible: 2129.370800604054

Heurísticas aplicadas: ['limitar\_funcion\_objetivo']

No se encontró una solución óptima, la siguiente es la mejor solución encontrada:

Distancia total recorrida: 1461.9020286291757

#### Sin Heurística:

Tiempo de ejecución: 00:40

Distancia mínima entre nodos: 0.6403124237432862

Distancia máxima entre nodos: 234.3041826344549

Distancia promedio entre nodos: 100.75829712883072

Distancia Total mínima posible: 887.2378335850225

Distancia Total máxima posible: 2129.370800604054

Heurísticas aplicadas: []

No se encontró una solución óptima, la siguiente es la mejor solución encontrada:

Distancia total recorrida: 1697.8332135729727

¿Cuál es la diferencia entre los dos casos?

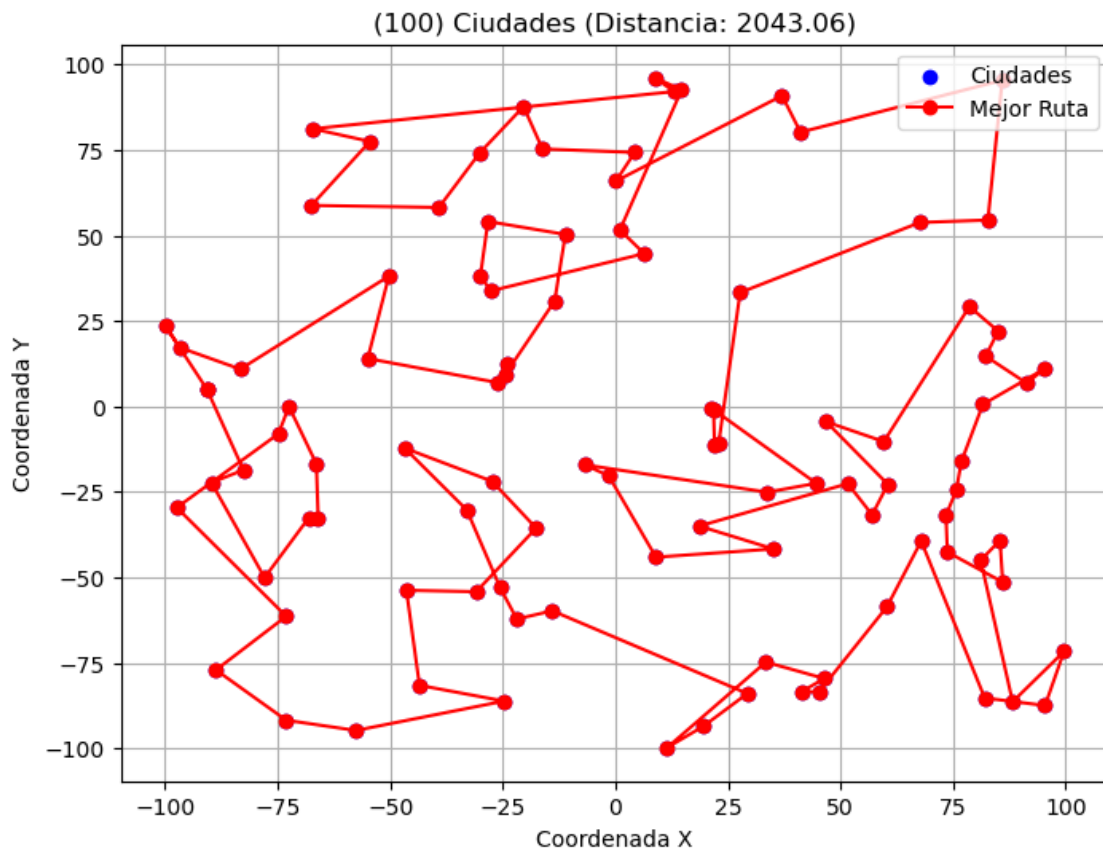
La aplicación de la heurística permitió encontrar una mejor solución (más corta) en el mismo tiempo de ejecución.

**¿Sirve esta heurística para cualquier caso? ¿Cuál pudiera ser una razón?**

La heurística `limitar_funcion_objetivo` puede ser útil en algunos casos, pero no garantiza mejoras en todos los contextos. Esto se debe a que:

1. Las heurísticas suelen estar diseñadas para reducir el espacio de búsqueda, enfocándose en soluciones más prometedoras, pero:
  2. Pueden omitir soluciones óptimas si las restricciones impuestas son demasiado fuertes.
- En este caso, funcionó bien porque probablemente acotó las soluciones malas, permitiendo al algoritmo concentrarse en rutas más eficientes.

#### D. Aplicar heurística de vecinos cercanos



**Con heurística:**

Tiempo de ejecución: 01:02  
Distancia mínima entre nodos: 0.6403124237432862  
Distancia máxima entre nodos: 245.61115609841502  
Distancia promedio entre nodos: 100.94053659793771  
Distancia Total mínima posible: 1269.7606127710123  
Distancia Total máxima posible: 3047.4254706504294  
Heurísticas aplicadas: ['vecino\_cercano']  
No se encontró una solución óptima, la siguiente es la mejor solución encontrada:  
Distancia total recorrida: 2043.0594118010144

### **Sin heurística:**

Tiempo de ejecución: 01:02  
Distancia mínima entre nodos: 0.6403124237432862  
Distancia máxima entre nodos: 245.61115609841502  
Distancia promedio entre nodos: 100.94053659793771  
Distancia Total mínima posible: 1269.7606127710123  
Distancia Total máxima posible: 3047.4254706504294  
Heurísticas aplicadas: []  
No se encontró una solución óptima, la siguiente es la mejor solución encontrada:  
Distancia total recorrida: 0  
La ruta contiene valores nulos, no se encontró una solución válida.

### **¿Cuál es la diferencia entre los dos casos?**

Si bien es cierto que tanto los tiempos de ejecución y los valores estadísticos son los mismos, la diferencia principal es que cuando se aplicó la heurística del vecino\_cercano se pudo llegar a una ruta con distancia total (no óptima).

### **¿Sirve esta heurística para cualquier caso?**

La heurística vecina\_cercano no garantiza la mejor solución, pero sí ayuda a obtener una solución válida en muchos casos.

### **Conclusiones:**

- El TSP es un problema NP-duro, lo que significa que el tiempo de resolución crece exponencialmente con el número de ciudades.
- Para pocas ciudades (10 o 20), se puede analizar la mayoría del espacio de soluciones sin heurísticas.
- A partir de 30 ciudades, la cantidad de rutas posibles se vuelve muy difícil de cuantificar, por lo que los métodos exactos sin heurísticas fallan.
- Las heurísticas permiten encontrar soluciones viables en tiempos razonables
- Usar heurísticas aumentan la capacidad para resolver instancias con muchas ciudades



- Aunque no óptimas, las soluciones suelen ser suficientemente buenas para la mayoría de las aplicaciones prácticas.
- Hay diferentes tipos de heurísticas por lo que la calidad depende del tipo de heurística usada