



**Facultad Regional Rosario**  
Universidad Tecnológica Nacional

**Carrera:**

Ingeniería en Sistemas de Información

**Asignatura:**

Algoritmos Genéticos

**Comisión:**

3ek03

**Ciclo lectivo:**

2024

**Título:**

Trabajo Práctico N°3 – Problema del viajante de comercio

**Docentes:**

DIAZ, Daniela

LOMBARDO, Víctor

**Grupo Nro. 5:**

GALLEGOS, Nicolás Gabriel – Leg.: 51367 – nicogabrielgallegos@gmail.com



## Índice

Índice.....	2
Enunciado .....	3
Herramientas de programación .....	4
Metodología de desarrollo abordada.....	5
Búsqueda Exhaustiva.....	5
Búsqueda Heurística .....	5
Algoritmo Genético.....	6
Código del programa principal.....	8
Resultados 2 - A (Búsqueda Heurística) .....	14
Resultados 2 - B (Mínimo por Búsqueda Heurística).....	27
Resultados 2 - C (Algoritmo Genético) .....	28
Conclusiones.....	29



## Enunciado

Sean  $N$  ciudades de un territorio. La distancia entre cada ciudad viene dada por la matriz  $D: N \times N$ , donde  $d[x, y]$  representa la distancia que hay entre la ciudad  $x$  y la ciudad  $y$ .

El objetivo es encontrar una ruta que, comenzando y terminando en una ciudad concreta, pase una sola vez por cada una de las ciudades y minimice la distancia recorrida por el viajante.

Ejercicios:

- 1) Hallar la ruta de distancia mínima que logre unir todas las capitales de provincias de la República Argentina, utilizando un método exhaustivo. ¿Puede resolver el problema? Justificar de manera teórica.
- 2) Realizar un programa que cuente con un menú con las siguientes opciones:
  - a) Permitir ingresar una provincia, hallar la ruta de distancia mínima que logre unir todas las capitales de provincias de la República Argentina partiendo de dicha capital utilizando la siguiente heurística: “Desde cada ciudad, ir a la ciudad más cercana no visitada”. Recordar regresar siempre a la ciudad de partida. Presentar un mapa de la República Argentina con el recorrido indicado. Además, indicar la ciudad de partida, el recorrido completo y la longitud del trayecto. El programa deberá permitir seleccionar la capital que el usuario desee ingresar como inicio del recorrido.
  - b) Encontrar el recorrido mínimo para visitar todas las capitales de provincias de la República Argentina siguiendo la heurística mencionada en el punto a. Deberá mostrar como salida el recorrido y la longitud del trayecto.
  - c) Hallar la ruta de distancia mínima que logre unir todas las capitales de provincias de la República Argentina, utilizando un algoritmo genético.



## Herramientas de programación

Para abordar el problema propuesto, se utilizará el lenguaje de programación Python, en su versión 3.12.2 64-bits.

El motivo de esta decisión es la facilidad y flexibilidad de dicho lenguaje sobre otros como por ejemplo C++, C# o Java; ya que Python tiene tanto una sintaxis como un funcionamiento menos restrictivo que los demás lenguajes mencionados, lo que resulta más práctico a la hora de desarrollar una solución intuitiva y buena del problema de la Mochila.

Para este trabajo se optó por utilizar la extensión Jupyter Notebooks de Visual Studio Code para poder dividir el código en distintas celdas y ordenarlo de una mejor forma en un único archivo de extensión ipynb.

El archivo ipynb del programa y su versión exportada a py se encuentran en el repositorio [AG – TP / 3er Trabajo Práctico](#) de GitHub.

Además, se utilizan las librerías tkinter, para diseñar la interfaz de la aplicación; y matplotlib, para mostrar el mapa de la República Argentina junto con los recorridos hallados.



## Metodología de desarrollo abordada

### Búsqueda Exhaustiva

El problema del viajante de comercio, o por sus siglas en inglés “TSP”, se encuentra dentro del conjunto de problemas llamados “NP-Complejos”. Este tipo de problemas no pueden resolverse en un tiempo polinómico en función de los datos de entrada.

El tiempo de resolución del TSP es  $N!$ , donde  $N$  es el número de provincias a recorrer. Esto hace que intentar recorrer el espacio de soluciones del problema sea muy complicado y costoso incluso para valores de  $N$  no tan grandes.

Por ejemplo, asumiendo que una computadora calcula la longitud de cada ciclo en un microsegundo, tardaría algo más de 3 segundos en resolver el problema para 10 ciudades, algo más de medio minuto para 11 ciudades y 77146 años en resolverlo para solo 20 ciudades.

Teniendo en cuenta que nuestro problema cuenta con 24 ciudades, no existe la posibilidad de poder resolverlo a través de una única búsqueda exhaustiva debido al requerimiento del tiempo, aunque existen otras alternativas.

Un ejemplo podría ser dividir las 24 ciudades en dos grupos de 12 ciudades, o incluso en tres grupos de 8 ciudades. Luego, considerar a cada uno de estos grupos como problemas independientes y hallar soluciones parciales a través de distintas búsquedas exhaustivas más pequeñas. Y finalmente juntar estas soluciones parciales en una única solución que podría ser una buena aproximación a la solución real de haber realizado una búsqueda exhaustiva que abarque todo el problema. De todas maneras, esto obviamente no nos asegura encontrar la solución óptima al problema.

### Búsqueda Heurística

Este acercamiento al problema facilita el hecho de encontrar una buena solución en un período de tiempo mucho más razonable que el de la búsqueda exhaustiva.

En este caso, como se solicita en el enunciado, se utilizará la heurística: “*Desde cada ciudad ir a la ciudad más cercana no visitada*”, donde existe una matriz  $D$  de dimensiones  $N \times N$  (En nuestro caso  $24 \times 24$ ) cuyos elementos  $d_{ij}$  indican la distancia entre las ciudades  $i$  y  $j$  para  $i, j = 1, 2, 3, \dots, 24$ .



El algoritmo en cada paso explora la fila o columna de la matriz  $D$  que corresponda con la ciudad en la que se encuentre actualmente y busca la ciudad más cercana. Luego, la agrega a la solución y actualiza la ciudad en la que se encuentra para ir al siguiente paso. Este proceso se repite hasta haber recorrido las 24 provincias.

## Algoritmo Genético

Otro método para conseguir una buena solución para el TSP es utilizar algoritmos genéticos.

Se considerarán poblaciones de 50 cromosomas, donde cada cromosoma será una permutación de 24 números enteros del 0 al 23. Cada uno de estos genes representa a una ciudad. El cromosoma  $i$  se representa como  $c_i = (c_{i_1}, c_{i_2}, c_{i_3}, \dots, c_{i_{23}}, c_{i_{24}})$ .

En este problema, a diferencia de otros problemas tratados anteriormente, el orden de los elementos que conforman la solución, si es un detalle a tener en cuenta ya que modifica la solución. Pero como esta representa un recorrido cerrado, “rotar” o “desplazar” todos los elementos de la solución  $n$  lugares hacia la izquierda o la derecha (con  $n$  siendo un número entero), no modifican la solución.

Debido a esto, durante la ejecución del algoritmo genético se podrían generar soluciones que parezcan diferentes, pero que no lo son. Por ejemplo, tomando solo 4 ciudades, las soluciones  $(0, 1, 2, 3)$ ,  $(1, 2, 3, 0)$ ,  $(2, 3, 0, 1)$  y  $(3, 0, 1, 2)$  hacen referencia al mismo recorrido, siendo la única diferencia su ciudad de partida (que no afecta a la longitud del recorrido).

Por eso, para evitar que esto suceda e intentar generar más variedad en los cromosomas, el primer gen de cada cromosoma será siempre 0, mientras que los otros 23 podrán variar entre los números naturales del 1 al 23. Esto asegura que dos cromosomas que sean distintos hagan referencia a recorridos distintos.

Luego, de ser necesario modificar la ciudad de partida de un determinado recorrido, la solución puede ser “rotada” tantos lugares como haga falta.

Lo siguiente es elegir los parámetros y funciones que se tendrán en cuenta para el algoritmo.

Para comenzar, una buena forma de definir el fitness de un cromosoma dentro de este problema sería calculando el recíproco de la longitud de su recorrido, ya que mientras menor sea la longitud, mayor será su fitness.



Sin embargo, con el fin de evitar tener que usar el operador de división para calcular el fitness de cada uno de los individuos por ser una operación costosa, se definirá al fitness del cromosoma  $c_i$  como el opuesto de la longitud de su recorrido:

$$fitness(c_i) = -longitud(c_i) = -\left(\sum_{k=1}^{23} (d[c_i[k], c_i[k+1]]) + d[c_i[24], c_i[1]]\right)$$

Así, por ejemplo, siendo  $c_1$  y  $c_2$  dos cromosomas cuyas longitudes son 10000 km y 12000 km respectivamente, se puede verificar que  $fitness(c_1) > fitness(c_2)$

Luego, se utilizará el método de selección de torneo, que elegirá de manera aleatoria al 40% de la población (20 individuos) y seleccionará finalmente al cromosoma de mayor fitness, es decir, al cromosoma cuyo recorrido tenga la menor distancia.

Las probabilidades de crossover y mutación serán 90% y 20% respectivamente. Se utilizará un crossover cíclico para asegurar que un mismo cromosoma no pueda tener ciudades repetidas, ni pueda variar su cantidad de genes. La mutación de un cromosoma se hará intercambiando dos genes aleatorios del mismo.



## Código del programa principal

A continuación, se encuentran los fragmentos de código más importantes para la resolución de los problemas planteados. Sin embargo, el código completo del programa se encuentra subido en el repositorio [AG – TP / 3er Trabajo Práctico](#) de GitHub.

```
CANT_CIUDES = 24

NOMBRES_CIUDES = ["Ciudad Autónoma de Buenos Aires", "Córdoba", "Corrientes", "Formosa",
                  "La Plata", "La Rioja", "Mendoza", "Neuquén", "Paraná", "Posadas", "Rawson",
                  "Resistencia", "Río Gallegos", "San Fernando del Valle de Catamarca",
                  "San Miguel de Tucumán", "San Salvador de Jujuy", "Salta", "San Juan", "San Luis",
                  "Santa Fe", "Santa Rosa", "Santiago del Estero", "Ushuaia", "Viedma"]

DISTANCIAS = [[ 646], # 1 - Córdoba
               [ 792, 677], # 2 - Corrientes
               [ 933, 824, 157], # 3 - Formosa
               [ 53, 698, 830, 968], # 4 - La Plata
               [ 986, 340, 814, 927, 1038], # 5 - La Rioja
               [ 985, 466, 1131, 1269, 1029, 427], # 6 - Mendoza
               [ 989, 907, 1534, 1690, 1005, 1063, 676], # 7 - Neuquén
               [ 375, 348, 580, 656, 427, 659, 790, 1053], # 8 - Paraná
               [ 834, 919, 291, 263, 857, 1098, 1384, 1709, 658], # 9 - Posadas
               [1127, 1321, 1845, 1999, 1116, 1548, 1201, 543, 1345, 1951], # 10 - Rawson
               [ 794, 669, 13, 161, 833, 802, 1121, 1529, 498, 305, 1843], # 11 - Resistencia
               [2082, 2281, 2819, 2974, 2064, 2473, 2081, 1410, 2320, 2914, 975, 2818], # 12 - Río Gallegos
               [ 979, 362, 691, 793, 1030, 149, 569, 1182, 622, 980, 1647, 678, 2587], # 13 - San Fermín del Valle de Catamarca
               [1080, 517, 633, 703, 1132, 330, 756, 1370, 707, 924, 1827, 620, 2773, 189], # 14 - San Miguel de Tucumán
               [1334, 809, 742, 750, 1385, 600, 1023, 1658, 959, 1007, 2120, 729, 3063, 477, 293], # 15 - San Salvador de Jujuy
               [1282, 745, 719, 741, 1333, 533, 957, 1591, 906, 992, 2054, 706, 2997, 410, 228, 67], # 16 - Salta
               [1005, 412, 1039, 1169, 1053, 283, 152, 824, 757, 1306, 1340, 1020, 2331, 430, 612, 874, 808], # 17 - San Juan
               [ 749, 293, 909, 1117, 795, 435, 235, 643, 574, 1200, 1113, 961, 2046, 540, 727, 1817, 950, 2041], # 18 - San Luis
               [ 303, 330, 498, 654, 444, 640, 775, 1049, 19, 664, 1349, 495, 2325, 602, 689, 942, 889, 740, 560], # 19 - Santa Fe
               [ 579, 577, 1136, 1293, 602, 834, 586, 422, 642, 1293, 745, 1132, 1712, 915, 1088, 1382, 1316, 686, 412, 641], # 20 - Santa Rosa
               [ 939, 401, 535, 629, 991, 311, 713, 1286, 566, 827, 1721, 523, 2677, 166, 141, 414, 353, 583, 643, 547, 977], # 21 - Santiago del Estero
               [2373, 2618, 3131, 3284, 2350, 2821, 2435, 1762, 2635, 3207, 1300, 3130, 359, 2931, 3116, 3408, 3341, 2585, 2392, 2641, 2044, 3016], # 22 - Ushuaia
               [ 799, 1047, 1527, 1681, 789, 1311, 1019, 479, 1030, 1624, 327, 1526, 1294, 1391, 1562, 1855, 1790, 1141, 882, 1035, 477, 1446, 1605], # 23 - Viedma
               [0], # 24 - (no se usa)]
```

```
def distancia(partida: int, destino: int) -> int:
    """
    Recibe como parámetros una ciudad de partida y otra de destino.
    Devuelve la distancia en línea recta entre las mismas.
    """
    if partida == destino or partida < 0 or partida > 23 or destino < 0 or destino > 23:
        return 0

    if partida > destino:
        return DISTANCIAS[partida - 1][destino]
    else:
        return DISTANCIAS[destino - 1][partida]

def ciudad_mas_cercana(partida: int, ciudades: list[int], mascara: list[int] | None = None) -> int | None:
    """
    Recibe como parámetros una ciudad de partida, una lista de ciudades y una máscara de bits.
    Devuelve la ciudad dentro del conjunto que tenga la menor distancia con la ciudad de partida y esté incluida en la máscara.
    """
    mascara = validar_mascara(mascara, len(ciudades))

    d_min = float("inf")
    c_min = None
    for i in range(len(ciudades)):
        if mascara[i] == 0 and ciudades[i] != partida:
            d = distancia(partida, ciudades[i])
            if d < d_min:
                d_min = d
                c_min = ciudades[i]
    return c_min

def longitud_recorrido(recorrido: list[int]) -> int:
    """
    Recibe como parámetro una lista de ciudades.
    Devuelve la distancia total del recorrido que pasa por esas ciudades.
    """
    l = 0
    for i in range(-1, len(recorrido) - 1):
        l += distancia(recorrido[i], recorrido[i+1])
    return l
```





```
def validar_mascara(mascara: list[int] | None, n: int) -> list[int]:  
    """  
    Recibe como parámetro una máscara y el tamaño de la lista que enmascara.  
  
    Si el tamaño de la máscara no coincide con el tamaño de la lista,  
    crea una nueva máscara de igual tamaño que la lista. Sino, la máscara  
  
    Devuelve una máscara de bits corregida  
    """  
    if mascara is None or len(mascara) != n:  
        mascara = [0 for i in range(n)]  
    return mascara  
  
def desplazar_lista(lista: list, n: int) -> None:  
    """  
    Recibe como parámetro una lista.  
  
    Desplaza los elementos de la lista una cantidad n de espacios hacia la izquierda.  
    """  
    nueva_lista = []  
    for i in range(len(lista)):  
        nueva_lista.append(lista[(i + n) % len(lista)])  
  
    for i in range(len(lista)):  
        lista[i] = nueva_lista[i]
```

```
def recorrido_heuristico(partida: int, ciudades: list[int], mascara: list[int] | None = None) -> list[int]:  
    """  
    Recibe como parámetro una ciudad de partida, una lista de ciudades y una máscara de bits.  
  
    Devuelve una lista de ciudades que representa el camino a realizar para que la distancia del recorrido entre dichas ciudades sea mínimo.  
    """  
    mascara = validar_mascara(mascara, len(ciudades))  
  
    recorrido = [partida]  
    mascara[partida] = 1  
    for i in range(mascara.count(0)):  
        siguiente_ciudad: int = ciudad_mas_cercana(recorrido[i], ciudades, mascara)  
        recorrido.append(siguiente_ciudad)  
        mascara[siguiente_ciudad] = 1  
    recorrido.append(partida)  
    return recorrido  
  
def recorrido_minimo_heuristico(partida: int, ciudades: list[int], mascara: list[int] | None = None) -> list[int]:  
    aux = [i for i in mascara]  
    recorrido = recorrido_heuristico(partida, ciudades, mascara)  
    for i in range(len(aux)):  
        if aux[i] == 0:  
            mascara = [i for i in aux]  
            r_aux = recorrido_heuristico(i, ciudades, mascara)  
            if longitud_recorrido(r_aux) < longitud_recorrido(recorrido):  
                recorrido = r_aux  
    recorrido.pop(-1)  
    desplazar_lista(recorrido, recorrido.index(partida))  
    recorrido.append(partida)  
    return recorrido
```

```
def generar_cromosoma(ciudades: list[int]) -> list[int]:  
    c: list[int] = [i for i in ciudades]  
    cromosoma: list[int] = [c.pop(0)]  
    for i in range(len(c)):  
        r = random.randint(0, len(c) - 1)  
        cromosoma.append(c.pop(r))  
    return cromosoma  
  
def generar_poblacion(n: int, ciudades: list[int]) -> list[list[int]]:  
    poblacion: list[list[int]] = []  
    for i in range(n):  
        poblacion.append(generar_cromosoma(ciudades))  
    return poblacion  
  
def fitness(cromosoma: list[int]) -> float:  
    return -longitud_recorrido(cromosoma)
```



```
def seleccion(poblacion: list[list[int]], porcentaje_torneo: float) -> list[int]:
    indices_disponibles: list[int] = [i for i in range(len(poblacion))]
    cromosomas_seleccionados: list[int] = []

    for i in range(math.ceil(len(poblacion) * porcentaje_torneo)):
        indice_aleatorio: int = random.randint(0, len(indices_disponibles) - 1)
        cromosomas_seleccionados.append(poblacion[indices_disponibles.pop(indice_aleatorio)])

    mejor_cromosoma: int = cromosomas_seleccionados[0]

    for c in cromosomas_seleccionados:
        if fitness(c) > fitness(mejor_cromosoma):
            mejor_cromosoma = c

    return mejor_cromosoma

def crossover_ciclico(padre: list[int], madre: list[int], probabilidad: int = 0.9) -> tuple[list[int], list[int]]:
    if random.random() < probabilidad:
        lista_indices: list[int] = []
        indice_nuevo = random.randint(1, len(padre) - 1)
        lista_indices.append(indice_inicial)

        while True:
            indice_nuevo = padre.index(madre[indice_nuevo])
            if indice_nuevo == indice_inicial:
                break
            lista_indices.append(indice_nuevo)

        hijo_1 = [i for i in padre]
        hijo_2 = [i for i in madre]

        for i in lista_indices:
            aux = hijo_1[i]
            hijo_1[i] = hijo_2[i]
            hijo_2[i] = aux
    else:
        hijo_1 = [i for i in padre]
        hijo_2 = [i for i in madre]
    return (hijo_1, hijo_2)

def mutacion(cromosoma: list[int], genes: int = 2, probabilidad: int = 0.2) -> None:
    if random.random() < probabilidad:
        indices: list[int] = [(i + 1) for i in range(len(cromosoma) - 1)]
        genes_mezclados: list[int] = []
        for i in range(genes):
            r = random.randint(1, len(indices) - 1)
            genes_mezclados.append(indices.pop(r))

        aux = cromosoma[genes_mezclados[0]]
        for i in range(len(genes_mezclados) - 1):
            cromosoma[genes_mezclados[i]] = cromosoma[genes_mezclados[i + 1]]
        cromosoma[genes_mezclados[-1]] = aux

def siguiente_generacion(poblacion: list[list[int]], porcentaje_torneo: float) -> list[list[int]]:
    nueva_poblacion: list[list[int]] = []

    for i in range(int(len(poblacion) / 2)):
        padre = seleccion(poblacion, porcentaje_torneo)
        madre = seleccion(poblacion, porcentaje_torneo)
        hijo_1, hijo_2 = crossover_ciclico(padre, madre)
        mutacion(hijo_1)
        mutacion(hijo_2)
        nueva_poblacion.append(hijo_1)
        nueva_poblacion.append(hijo_2)

    return nueva_poblacion
```



```
def recorrido_minimo_genetico(partida: int, ciudades: list[int], mascara: list[int], n: int = 50, m: int = 200, porcentaje_torneo: float = 0.4):

    validar_mascara(mascara, len(ciudades))
    c: list[int] = []
    for i in range(len(mascara)):
        if mascara[i] == 0:
            c.append(ciudades[i])

    poblacion = generar_poblacion(n, c)
    mejores_longitudes: list[int] = [min([longitud_recorrido(i) for i in poblacion])]
    peores_longitudes: list[int] = [max([longitud_recorrido(i) for i in poblacion])]
    promedios_longitudes: list[int] = [sum([longitud_recorrido(i) for i in poblacion]) / len(poblacion)]

    for i in range(m):
        poblacion = siguiente_generacion(poblacion, porcentaje_torneo)
        mejores_longitudes.append(min([longitud_recorrido(c) for c in poblacion]))
        peores_longitudes.append(max([longitud_recorrido(i) for i in poblacion]))
        promedios_longitudes.append(sum([longitud_recorrido(i) for i in poblacion]) / len(poblacion))

    mostrar_evolucion_poblaciones(mejores_longitudes, peores_longitudes, promedios_longitudes)

    recorrido = poblacion[0]
    for cromosoma in poblacion:
        if longitud_recorrido(cromosoma) < longitud_recorrido(recorrido):
            recorrido = cromosoma

    desplazar_lista(recorrido, recorrido.index(partida))
    recorrido.append(partida)
    return recorrido
```

```
def generar_recorrido(tipo: str = "Heurístico") -> None:
    mascara = []
    for i in range(len(ciudades_incluidas)):
        mascara.append(ciudades_incluidas[i].get())
    mascara[ciudad_partida.get()] = 0

    ciudades = [i for i in range(CANT_CIUDES)]

    if tipo == "Heurístico":
        r = recorrido_heuristico(ciudad_partida.get(), ciudades, mascara)
    elif tipo == "Mínimo":
        r = recorrido_minimo_heuristico(ciudad_partida.get(), ciudades, mascara)
    elif tipo == "Genético":
        r = recorrido_minimo_genetico(ciudad_partida.get(), ciudades, mascara)
    r = [CIUDES[i] for i in r]
    mostrar_recorrido_ttk(txt_recorrido, r)
    if chk_map.get() == 1:
        mostrar_mapa(r)

root = tk.Tk()
root.geometry("1020x650")
root.title("Problema del Viajante - República Argentina")

frm = ttk.Frame(root)
frm.grid()

lbl_frm_0 = ttk.LabelFrame(frm, text="Ciudad de partida")
lbl_frm_0.grid(column=0, row=0, padx=16, pady=16)

lbl_frm_1 = ttk.LabelFrame(frm, text="Ciudades incluidas en el recorrido")
lbl_frm_1.grid(column=1, row=0, padx=16, pady=16)

lbl_frm_2 = ttk.LabelFrame(frm, text="Resolver")
lbl_frm_2.grid(column=2, row=0, padx=16, pady=16)

ciudades_incluidas = [tk.IntVar() for i in range(CANT_CIUDES)]
ciudad_partida = tk.IntVar()

for i in range(CANT_CIUDES):
    radbtn = ttk.Radiobutton(lbl_frm_0, text=f"{NOMBRES_CIUDES[i]}", variable=ciudad_partida, value=i)
    chkbtn = ttk.Checkbutton(lbl_frm_1, text=f"{NOMBRES_CIUDES[i]}", variable=ciudades_incluidas[i], onvalue=0, offvalue=1)

    radbtn.grid(column=0, row=i+1, padx=2, pady=2, sticky="W")
    chkbtn.grid(column=1, row=i+1, padx=2, pady=2, sticky="W")

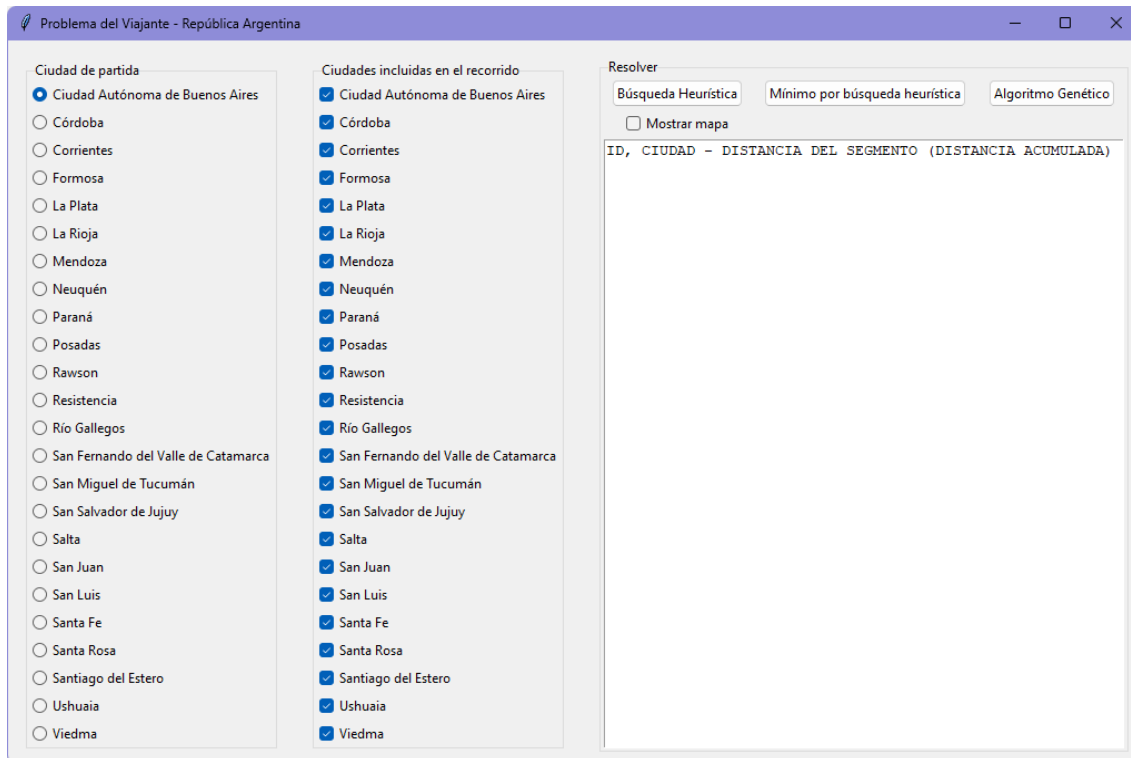
chk_map = tk.IntVar()

ttk.Button(lbl_frm_2, text="Búsqueda Heurística", command=lambda: generar_recorrido("Heurístico")).grid(column=0, row=0, padx=2, pady=2)
ttk.Button(lbl_frm_2, text="Mínimo por búsqueda heurística", command=lambda: generar_recorrido("Mínimo")).grid(column=1, row=0, padx=2, pady=2)
ttk.Button(lbl_frm_2, text="Algoritmo Genético", command=lambda: generar_recorrido("Genético")).grid(column=2, row=0, padx=2, pady=2)
ttk.Checkbutton(lbl_frm_2, text="Mostrar mapa", variable=chk_map, onvalue=1, offvalue=0).grid(column=0, row=1, padx=2, pady=2)
txt_recorrido = tk.Text(lbl_frm_2, height=34, width=58)
txt_recorrido.grid(column=0, row=3, columnspan=3, padx=2, pady=2, sticky="W")
txt_recorrido.insert(tk.END, f"ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)")

root.mainloop()
```



Al ejecutar el programa, se abrirá la siguiente ventana:



La primera columna consta de un grupo de opciones seleccionables, que permiten al usuario elegir la ciudad de partida del recorrido.

La segunda columna cuenta con las mismas ciudades, pero esta vez se trata de un grupo de selección múltiple, estas opciones indican qué ciudades estarán incluidas en el recorrido. Los resultados de este informe fueron obtenidos incluyendo todas las ciudades dentro del recorrido, esta es la opción por defecto del programa.

En la tercera columna se encuentra un panel con tres botones, cada uno de ellos resuelve un inciso del ejercicio número 2. Se puede marcar la opción de mostrar el mapa al terminar la búsqueda si el usuario lo desea. Debajo se puede ver una caja de texto, esta contendrá toda la información sobre los recorridos obtenidos luego de cada búsqueda realizada.



Luego de seleccionar una ciudad de partida, un conjunto de ciudades incluidas en el recorrido y presionar, por ejemplo, el botón “Algoritmo Genético”: el conjunto de ventanas que verá el usuario será el siguiente:

Problema del Viajante - República Argentina

Ciudad de partida

- ☐ Ciudad Autónoma de Buenos Aires
- ☐ Córdoba
- ☐ Corrientes
- ☐ Formosa
- ☐ La Plata
- ☐ La Rioja
- ☐ Mendoza
- ☐ Neuquén
- ☐ Paraná
- ☐ Posadas
- ☐ Rawson
- ☐ Resistencia
- ☐ Río Gallegos
- ☐ San Fernando del Valle de Catamarca
- ☐ San Miguel de Tucumán
- ☐ San Salvador de Jujuy
- ☐ Salta
- ☐ San Juan
- ☐ San Luis
- ☒ Santa Fe
- ☐ Santa Rosa
- ☐ Santiago del Estero
- ☐ Ushuaia
- ☐ Viedma

Ciudades incluidas en el recorrido

- ☒ Ciudad Autónoma de Buenos Aires
- ☒ Córdoba
- ☒ Corrientes
- ☐ Formosa
- ☐ La Plata
- ☐ La Rioja
- ☐ Mendoza
- ☒ Neuquén
- ☒ Paraná
- ☐ Posadas
- ☐ Rawson
- ☐ Resistencia
- ☐ Río Gallegos
- ☒ San Fernando del Valle de Catamarca
- ☒ San Miguel de Tucumán
- ☒ San Salvador de Jujuy
- ☒ Salta
- ☐ San Juan
- ☐ San Luis
- ☒ Santa Fe
- ☐ Santa Rosa
- ☐ Santiago del Estero
- ☒ Ushuaia
- ☐ Viedma

Resolver

☒ Mostrar mapa

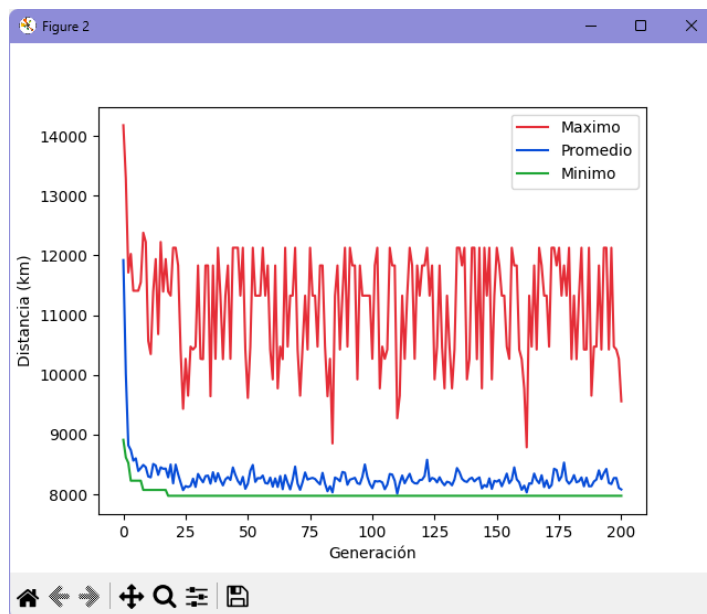
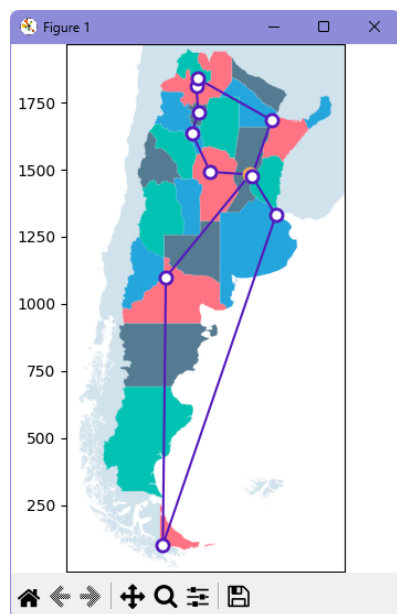
ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)

```

19, Santa Fe - 0 (0)
1, Córdoba - 330 (330)
13, San Fernando del Valle de Catamarca - 362 (692)
14, San Miguel de Tucumán - 189 (881)
16, Salta - 228 (1109)
15, San Salvador de Jujuy - 67 (1176)
2, Corrientes - 742 (1918)
8, Paraná - 500 (2418)
0, Ciudad Autónoma de Buenos Aires - 375 (2793)
22, Ushuaia - 2373 (5166)
7, Neuquén - 1762 (6928)
19, Santa Fe - 1049 (7977)

```

Distancia total recorrida: 7977





## Resultados 2 - A (Búsqueda Heurística)

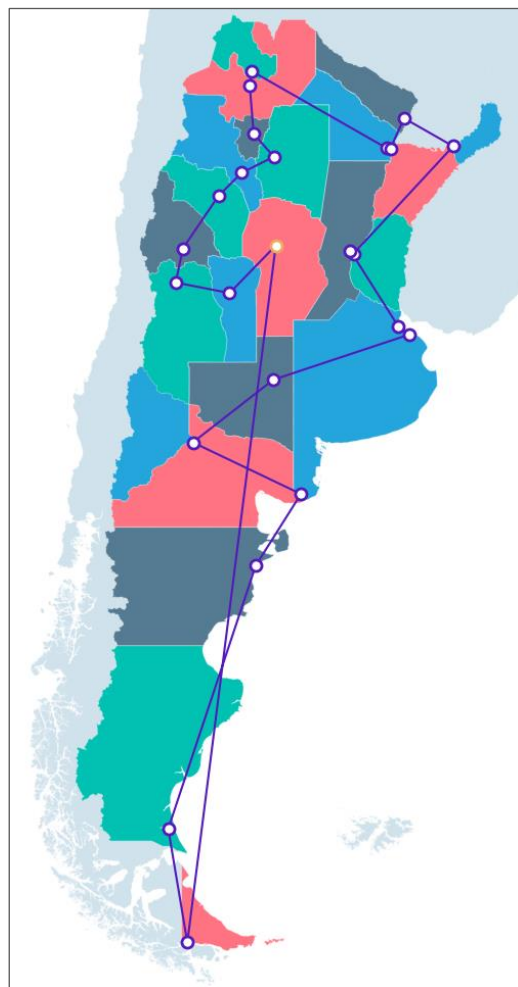
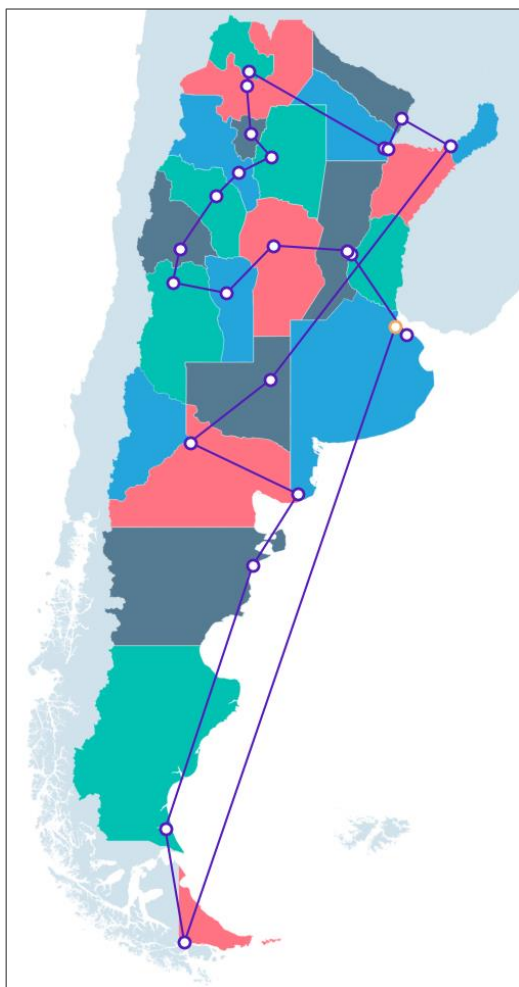
A continuación, se muestran los resultados tras realizar las búsquedas heurísticas

ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
0, Ciudad Autónoma de Buenos Aires - 0 (0)
4, La Plata - 53 (53)
8, Paraná - 427 (480)
19, Santa Fe - 19 (499)
1, Córdoba - 330 (829)
18, San Luis - 293 (1122)
6, Mendoza - 235 (1357)
17, San Juan - 152 (1509)
5, La Rioja - 283 (1792)
13, San Fernando del Valle de Catamarca - 149 (1941)
21, Santiago del Estero - 166 (2107)
14, San Miguel de Tucumán - 141 (2248)
16, Salta - 228 (2476)
15, San Salvador de Jujuy - 67 (2543)
11, Resistencia - 729 (3272)
2, Corrientes - 13 (3285)
3, Formosa - 157 (3442)
9, Posadas - 263 (3705)
20, Santa Rosa - 1293 (4998)
7, Neuquén - 422 (5420)
23, Viedma - 479 (5899)
10, Rawson - 327 (6226)
12, Río Gallegos - 975 (7201)
22, Ushuaia - 359 (7560)
0, Ciudad Autónoma de Buenos Aires - 2373 (9933)

Distancia total recorrida: 9933

ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
1, Córdoba - 0 (0)
18, San Luis - 293 (293)
6, Mendoza - 235 (528)
17, San Juan - 152 (680)
5, La Rioja - 283 (963)
13, San Fernando del Valle de Catamarca - 149 (1112)
21, Santiago del Estero - 166 (1278)
14, San Miguel de Tucumán - 141 (1419)
16, Salta - 228 (1647)
15, San Salvador de Jujuy - 67 (1714)
11, Resistencia - 729 (2443)
2, Corrientes - 13 (2456)
3, Formosa - 157 (2613)
9, Posadas - 263 (2876)
8, Paraná - 658 (3534)
19, Santa Fe - 19 (3553)
0, Ciudad Autónoma de Buenos Aires - 393 (3946)
4, La Plata - 53 (3999)
20, Santa Rosa - 602 (4601)
7, Neuquén - 422 (5023)
23, Viedma - 479 (5502)
10, Rawson - 327 (5829)
12, Río Gallegos - 975 (6804)
22, Ushuaia - 359 (7163)
1, Córdoba - 2618 (9781)

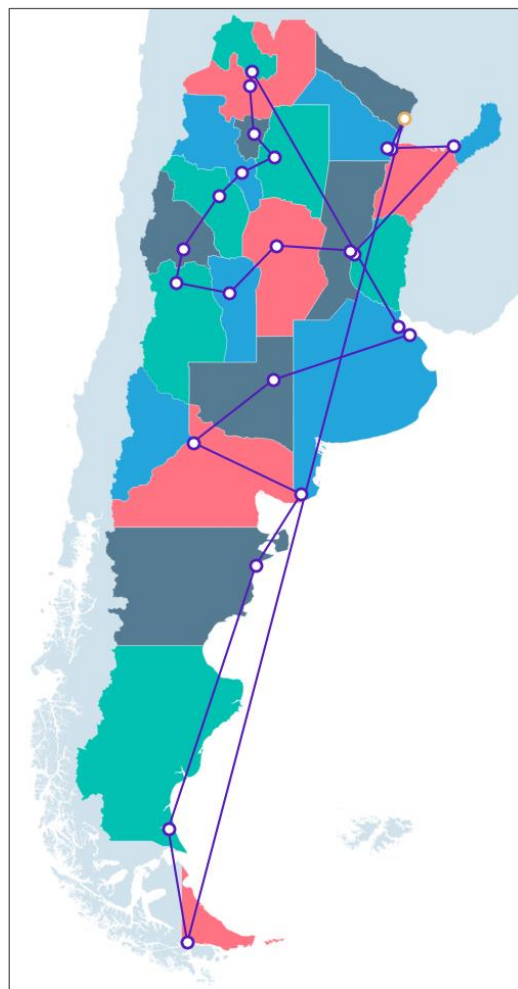
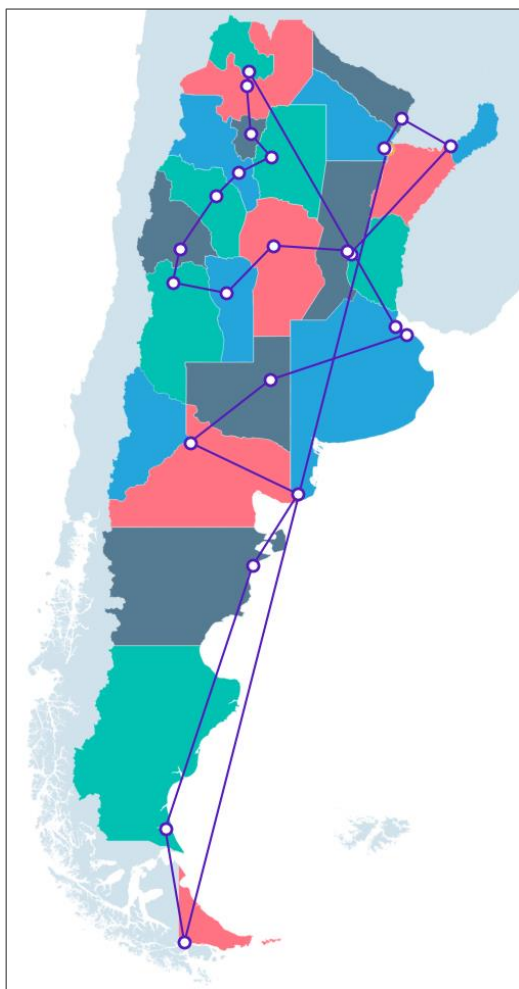
Distancia total recorrida: 9781





```
ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
2, Corrientes - 0 (0)
11, Resistencia - 13 (13)
3, Formosa - 161 (174)
9, Posadas - 263 (437)
8, Paraná - 658 (1095)
19, Santa Fe - 19 (1114)
1, Córdoba - 330 (1444)
18, San Luis - 293 (1737)
6, Mendoza - 235 (1972)
17, San Juan - 152 (2124)
5, La Rioja - 283 (2407)
13, San Fernando del Valle de Catamarca - 149 (2556)
21, Santiago del Estero - 166 (2722)
14, San Miguel de Tucumán - 141 (2863)
16, Salta - 228 (3091)
15, San Salvador de Jujuy - 67 (3158)
0, Ciudad Autónoma de Buenos Aires - 1334 (4492)
4, La Plata - 53 (4545)
20, Santa Rosa - 602 (5147)
7, Neuquén - 422 (5569)
23, Viedma - 479 (6048)
10, Rawson - 327 (6375)
12, Río Gallegos - 975 (7350)
22, Ushuaia - 359 (7709)
2, Corrientes - 3131 (10840)
Distancia total recorrida: 10840
```

```
ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
3, Formosa - 0 (0)
2, Corrientes - 157 (157)
11, Resistencia - 13 (170)
9, Posadas - 305 (475)
8, Paraná - 658 (1133)
19, Santa Fe - 19 (1152)
1, Córdoba - 330 (1482)
18, San Luis - 293 (1775)
6, Mendoza - 235 (2010)
17, San Juan - 152 (2162)
5, La Rioja - 283 (2445)
13, San Fernando del Valle de Catamarca - 149 (2594)
21, Santiago del Estero - 166 (2760)
14, San Miguel de Tucumán - 141 (2901)
16, Salta - 228 (3129)
15, San Salvador de Jujuy - 67 (3196)
0, Ciudad Autónoma de Buenos Aires - 1334 (4530)
4, La Plata - 53 (4583)
20, Santa Rosa - 602 (5185)
7, Neuquén - 422 (5607)
23, Viedma - 479 (6086)
10, Rawson - 327 (6413)
12, Río Gallegos - 975 (7388)
22, Ushuaia - 359 (7747)
3, Formosa - 3284 (11031)
Distancia total recorrida: 11031
```

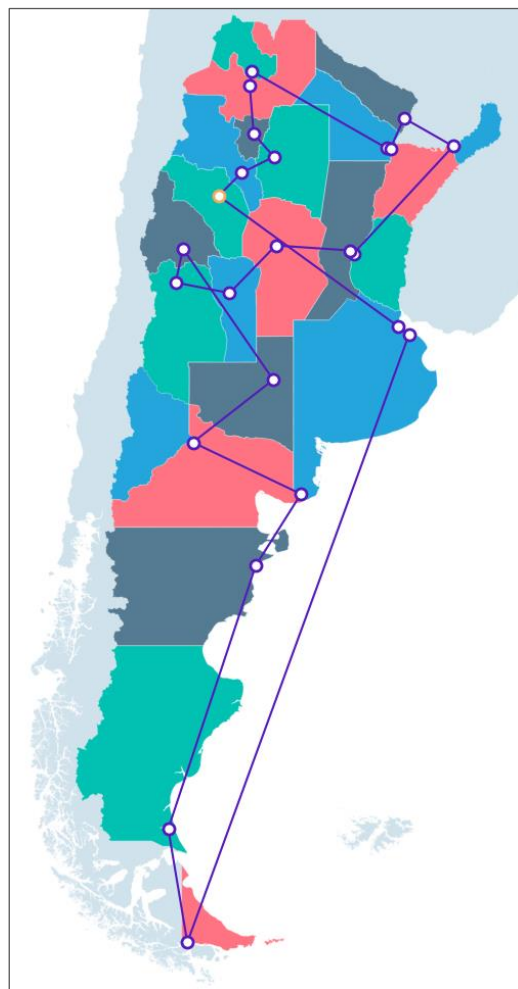
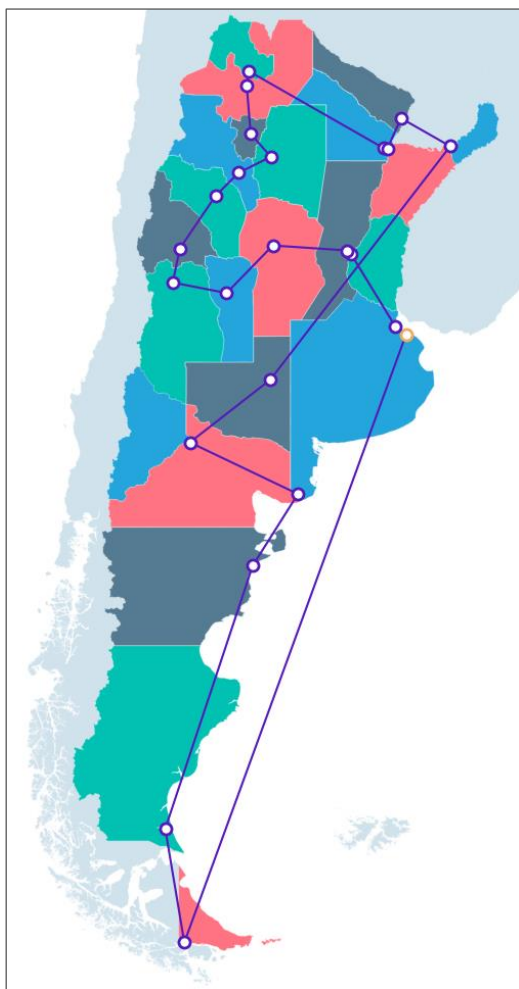






```
ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
4, La Plata - 0 (0)
0, Ciudad Autónoma de Buenos Aires - 53 (53)
8, Paraná - 375 (428)
19, Santa Fe - 19 (447)
1, Córdoba - 330 (777)
18, San Luis - 293 (1070)
6, Mendoza - 235 (1305)
17, San Juan - 152 (1457)
5, La Rioja - 283 (1740)
13, San Fernando del Valle de Catamarca - 149 (1889)
21, Santiago del Estero - 166 (2055)
14, San Miguel de Tucumán - 141 (2196)
16, Salta - 228 (2424)
15, San Salvador de Jujuy - 67 (2491)
11, Resistencia - 729 (3220)
2, Corrientes - 13 (3233)
3, Formosa - 157 (3390)
9, Posadas - 263 (3653)
20, Santa Rosa - 1293 (4946)
7, Neuquén - 422 (5368)
23, Viedma - 479 (5847)
10, Rawson - 327 (6174)
12, Río Gallegos - 975 (7149)
22, Ushuaia - 359 (7508)
4, La Plata - 2350 (9858)
Distancia total recorrida: 9858
```

```
ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
5, La Rioja - 0 (0)
13, San Fernando del Valle de Catamarca - 149 (149)
21, Santiago del Estero - 166 (315)
14, San Miguel de Tucumán - 141 (456)
16, Salta - 228 (684)
15, San Salvador de Jujuy - 67 (751)
11, Resistencia - 729 (1480)
2, Corrientes - 13 (1493)
3, Formosa - 157 (1650)
9, Posadas - 263 (1913)
8, Paraná - 658 (2571)
19, Santa Fe - 19 (2590)
1, Córdoba - 330 (2920)
18, San Luis - 293 (3213)
6, Mendoza - 235 (3448)
17, San Juan - 152 (3600)
20, Santa Rosa - 686 (4286)
7, Neuquén - 422 (4708)
23, Viedma - 479 (5187)
10, Rawson - 327 (5514)
12, Río Gallegos - 975 (6489)
22, Ushuaia - 359 (6848)
4, La Plata - 2350 (9198)
0, Ciudad Autónoma de Buenos Aires - 53 (9251)
5, La Rioja - 986 (10237)
Distancia total recorrida: 10237
```





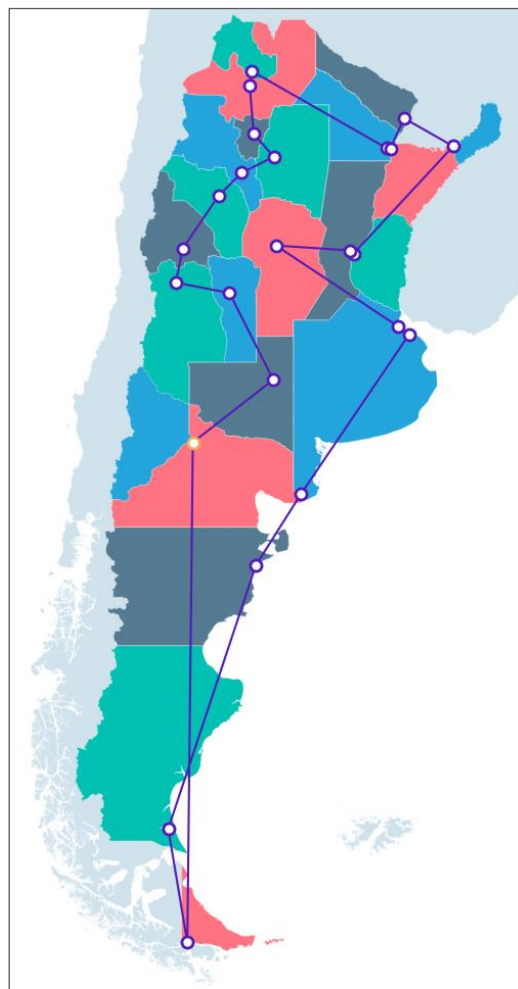
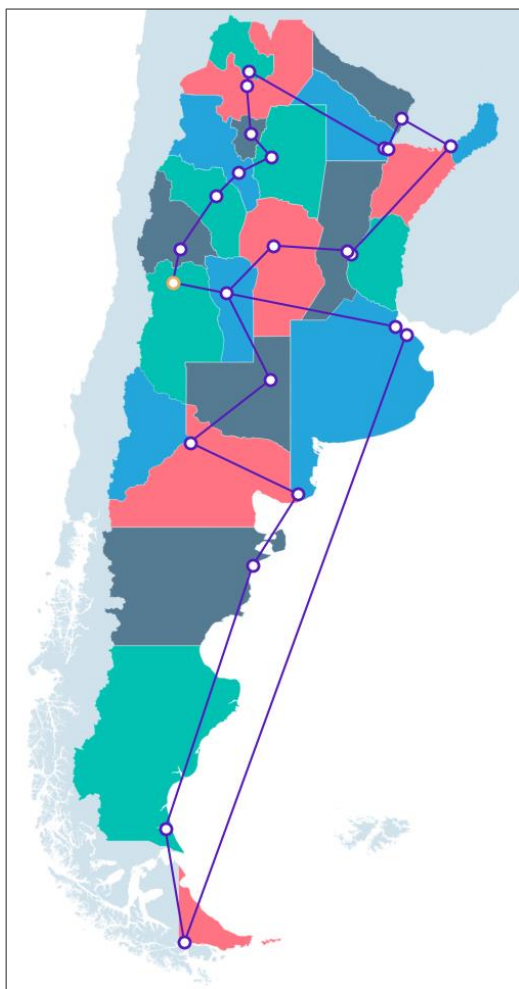


```
ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
6, Mendoza - 0 (0)
17, San Juan - 152 (152)
5, La Rioja - 283 (435)
13, San Fernando del Valle de Catamarca - 149 (584)
21, Santiago del Estero - 166 (750)
14, San Miguel de Tucumán - 141 (891)
16, Salta - 228 (1119)
15, San Salvador de Jujuy - 67 (1186)
11, Resistencia - 729 (1915)
2, Corrientes - 13 (1928)
3, Formosa - 157 (2085)
9, Posadas - 263 (2348)
8, Paraná - 658 (3006)
19, Santa Fe - 19 (3025)
1, Córdoba - 330 (3355)
18, San Luis - 293 (3648)
20, Santa Rosa - 412 (4060)
7, Neuquén - 422 (4482)
23, Viedma - 479 (4961)
10, Rawson - 327 (5288)
12, Río Gallegos - 975 (6263)
22, Ushuaia - 359 (6622)
4, La Plata - 2350 (8972)
0, Ciudad Autónoma de Buenos Aires - 53 (9025)
6, Mendoza - 985 (10010)

Distancia total recorrida: 10010
```

```
ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
7, Neuquén - 0 (0)
20, Santa Rosa - 422 (422)
18, San Luis - 412 (834)
6, Mendoza - 235 (1069)
17, San Juan - 152 (1221)
5, La Rioja - 283 (1504)
13, San Fernando del Valle de Catamarca - 149 (1653)
21, Santiago del Estero - 166 (1819)
14, San Miguel de Tucumán - 141 (1960)
16, Salta - 228 (2188)
15, San Salvador de Jujuy - 67 (2255)
11, Resistencia - 729 (2984)
2, Corrientes - 13 (2997)
3, Formosa - 157 (3154)
9, Posadas - 263 (3417)
8, Paraná - 658 (4075)
19, Santa Fe - 19 (4094)
1, Córdoba - 330 (4424)
0, Ciudad Autónoma de Buenos Aires - 646 (5070)
4, La Plata - 53 (5123)
23, Viedma - 789 (5912)
10, Rawson - 327 (6239)
12, Río Gallegos - 975 (7214)
22, Ushuaia - 359 (7573)
7, Neuquén - 1762 (9335)

Distancia total recorrida: 9335
```



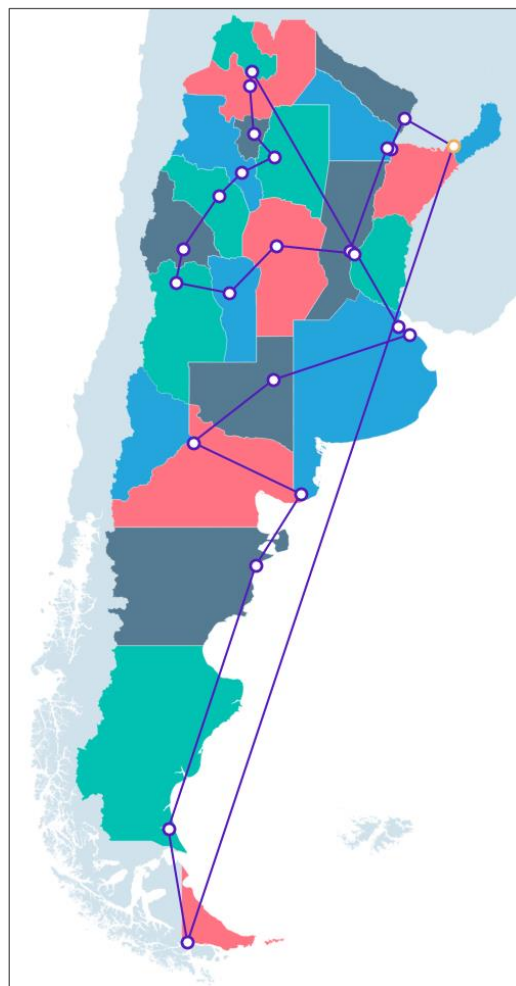
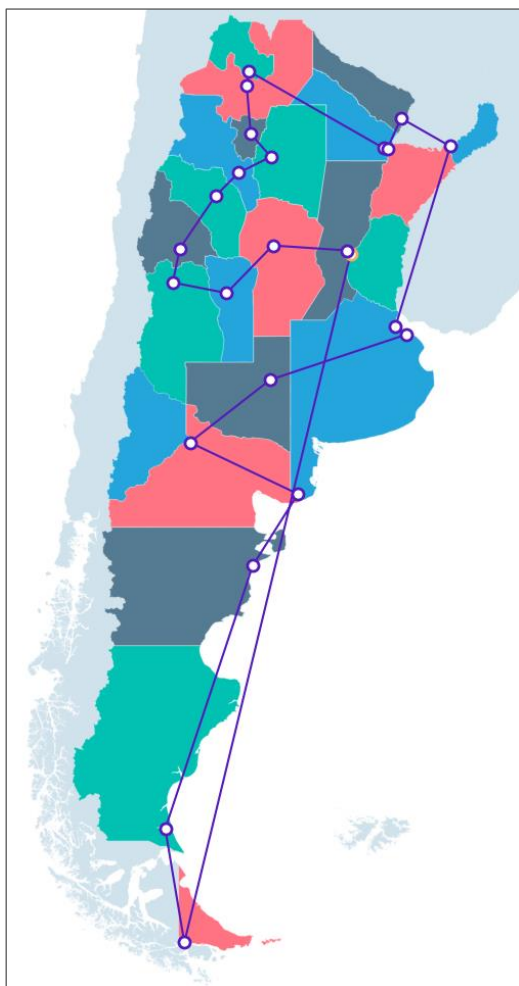


```
ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
8, Paraná - 0 (0)
19, Santa Fe - 19 (19)
1, Córdoba - 330 (349)
18, San Luis - 293 (642)
6, Mendoza - 235 (877)
17, San Juan - 152 (1029)
5, La Rioja - 283 (1312)
13, San Fernando del Valle de Catamarca - 149 (1461)
21, Santiago del Estero - 166 (1627)
14, San Miguel de Tucumán - 141 (1768)
16, Salta - 228 (1996)
15, San Salvador de Jujuy - 67 (2063)
11, Resistencia - 729 (2792)
2, Corrientes - 13 (2805)
3, Formosa - 157 (2962)
9, Posadas - 263 (3225)
0, Ciudad Autónoma de Buenos Aires - 834 (4059)
4, La Plata - 53 (4112)
20, Santa Rosa - 602 (4714)
7, Neuquén - 422 (5136)
23, Viedma - 479 (5615)
10, Rawson - 327 (5942)
12, Río Gallegos - 975 (6917)
22, Ushuaia - 359 (7276)
8, Paraná - 2635 (9911)

Distancia total recorrida: 9911
```

```
ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
9, Posadas - 0 (0)
3, Formosa - 263 (263)
2, Corrientes - 157 (420)
11, Resistencia - 13 (433)
19, Santa Fe - 495 (928)
8, Paraná - 19 (947)
1, Córdoba - 348 (1295)
18, San Luis - 293 (1588)
6, Mendoza - 235 (1823)
17, San Juan - 152 (1975)
5, La Rioja - 283 (2258)
13, San Fernando del Valle de Catamarca - 149 (2407)
21, Santiago del Estero - 166 (2573)
14, San Miguel de Tucumán - 141 (2714)
16, Salta - 228 (2942)
15, San Salvador de Jujuy - 67 (3009)
0, Ciudad Autónoma de Buenos Aires - 1334 (4343)
4, La Plata - 53 (4396)
20, Santa Rosa - 602 (4998)
7, Neuquén - 422 (5420)
23, Viedma - 479 (5899)
10, Rawson - 327 (6226)
12, Río Gallegos - 975 (7201)
22, Ushuaia - 359 (7560)
9, Posadas - 3207 (10767)

Distancia total recorrida: 10767
```



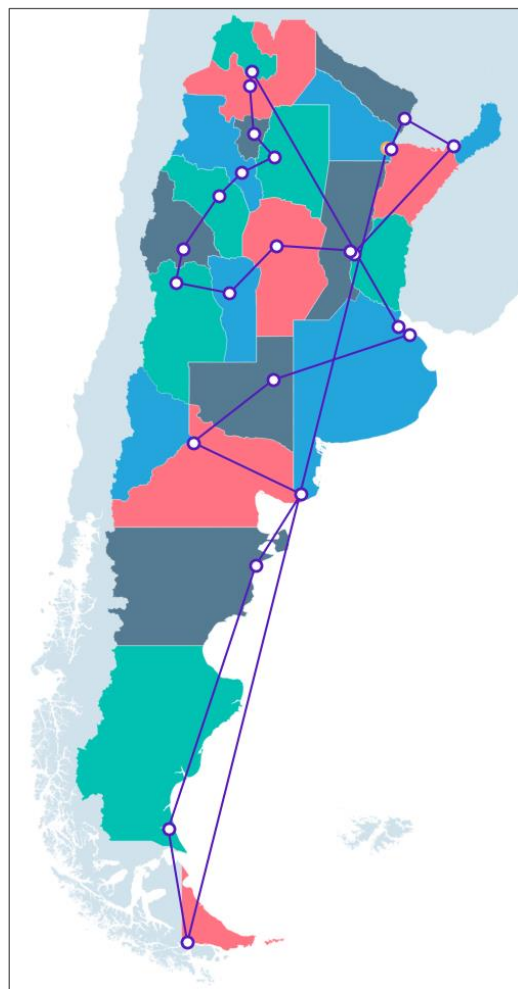
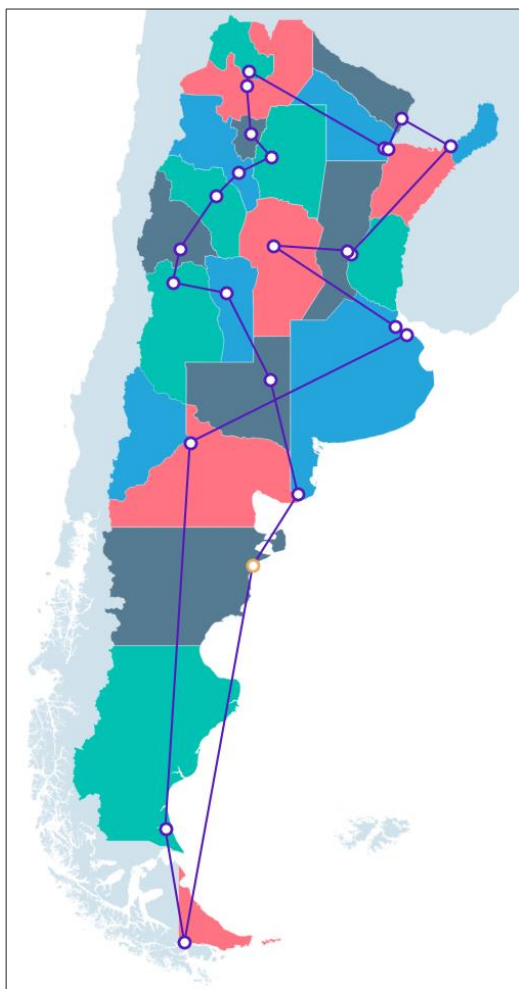


```
ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
10, Rawson - 0 (0)
23, Viedma - 327 (327)
20, Santa Rosa - 477 (804)
18, San Luis - 412 (1216)
6, Mendoza - 235 (1451)
17, San Juan - 152 (1603)
5, La Rioja - 283 (1886)
13, San Fernando del Valle de Catamarca - 149 (2035)
21, Santiago del Estero - 166 (2201)
14, San Miguel de Tucumán - 141 (2342)
16, Salta - 228 (2570)
15, San Salvador de Jujuy - 67 (2637)
11, Resistencia - 729 (3366)
2, Corrientes - 13 (3379)
3, Formosa - 157 (3536)
9, Posadas - 263 (3799)
8, Paraná - 658 (4457)
19, Santa Fe - 19 (4476)
1, Córdoba - 330 (4806)
0, Ciudad Autónoma de Buenos Aires - 646 (5452)
4, La Plata - 53 (5505)
7, Neuquén - 1005 (6510)
12, Río Gallegos - 1410 (7920)
22, Ushuaia - 359 (8279)
10, Rawson - 1300 (9579)

Distancia total recorrida: 9579
```

```
ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
11, Resistencia - 0 (0)
2, Corrientes - 13 (13)
3, Formosa - 157 (170)
9, Posadas - 263 (433)
8, Paraná - 658 (1091)
19, Santa Fe - 19 (1110)
1, Córdoba - 330 (1440)
18, San Luis - 293 (1733)
6, Mendoza - 235 (1968)
17, San Juan - 152 (2120)
5, La Rioja - 283 (2403)
13, San Fernando del Valle de Catamarca - 149 (2552)
21, Santiago del Estero - 166 (2718)
14, San Miguel de Tucumán - 141 (2859)
16, Salta - 228 (3087)
15, San Salvador de Jujuy - 67 (3154)
0, Ciudad Autónoma de Buenos Aires - 1334 (4488)
4, La Plata - 53 (4541)
20, Santa Rosa - 602 (5143)
7, Neuquén - 422 (5565)
23, Viedma - 479 (6044)
10, Rawson - 327 (6371)
12, Río Gallegos - 975 (7346)
22, Ushuaia - 359 (7705)
11, Resistencia - 3130 (10835)

Distancia total recorrida: 10835
```



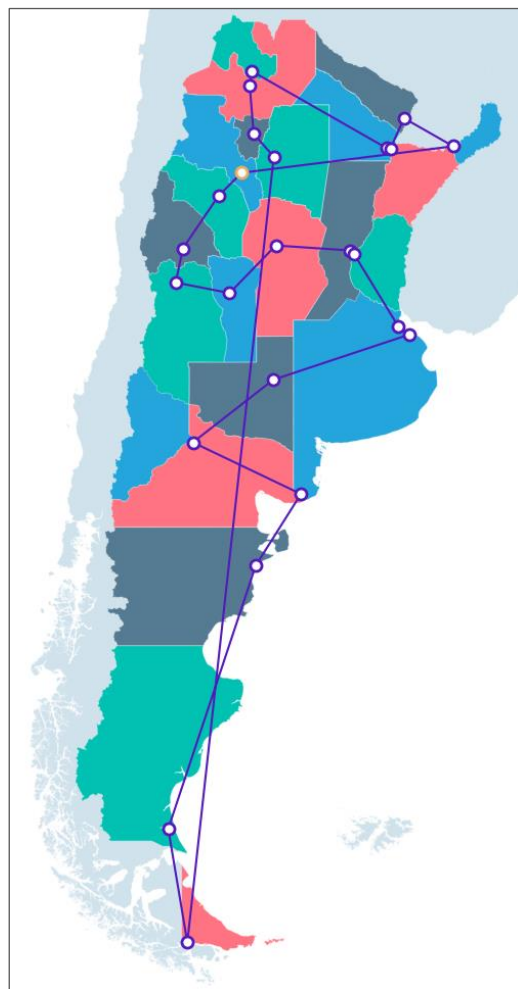
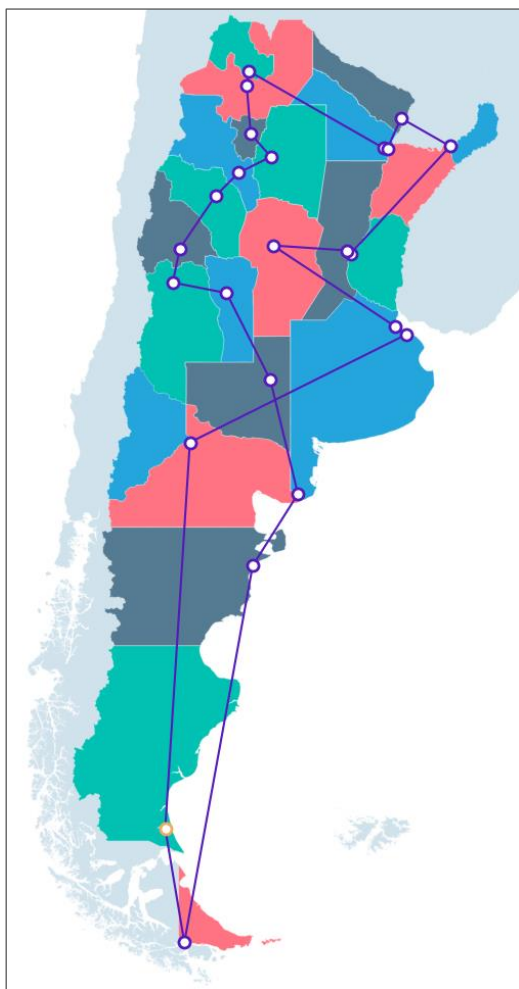


```
ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
12, Río Gallegos - 0 (0)
22, Ushuaia - 359 (359)
10, Rawson - 1300 (1659)
23, Viedma - 327 (1986)
20, Santa Rosa - 477 (2463)
18, San Luis - 412 (2875)
6, Mendoza - 235 (3110)
17, San Juan - 152 (3262)
5, La Rioja - 283 (3545)
13, San Fernando del Valle de Catamarca - 149 (3694)
21, Santiago del Estero - 166 (3860)
14, San Miguel de Tucumán - 141 (4001)
16, Salta - 228 (4229)
15, San Salvador de Jujuy - 67 (4296)
11, Resistencia - 729 (5025)
2, Corrientes - 13 (5038)
3, Formosa - 157 (5195)
9, Posadas - 263 (5458)
8, Paraná - 658 (6116)
19, Santa Fe - 19 (6135)
1, Córdoba - 330 (6465)
0, Ciudad Autónoma de Buenos Aires - 646 (7111)
4, La Plata - 53 (7164)
7, Neuquén - 1005 (8169)
12, Río Gallegos - 1410 (9579)

Distancia total recorrida: 9579
```

```
ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
13, San Fernando del Valle de Catamarca - 0 (0)
5, La Rioja - 149 (149)
17, San Juan - 283 (432)
6, Mendoza - 152 (584)
18, San Luis - 235 (819)
1, Córdoba - 293 (1112)
19, Santa Fe - 330 (1442)
8, Paraná - 19 (1461)
0, Ciudad Autónoma de Buenos Aires - 375 (1836)
4, La Plata - 53 (1889)
20, Santa Rosa - 602 (2491)
7, Neuquén - 422 (2913)
23, Viedma - 479 (3392)
10, Rawson - 327 (3719)
12, Río Gallegos - 975 (4694)
22, Ushuaia - 359 (5053)
21, Santiago del Estero - 3016 (8069)
14, San Miguel de Tucumán - 141 (8210)
16, Salta - 228 (8438)
15, San Salvador de Jujuy - 67 (8505)
11, Resistencia - 729 (9234)
2, Corrientes - 13 (9247)
3, Formosa - 157 (9404)
9, Posadas - 263 (9667)
13, San Fernando del Valle de Catamarca - 980 (10647)

Distancia total recorrida: 10647
```



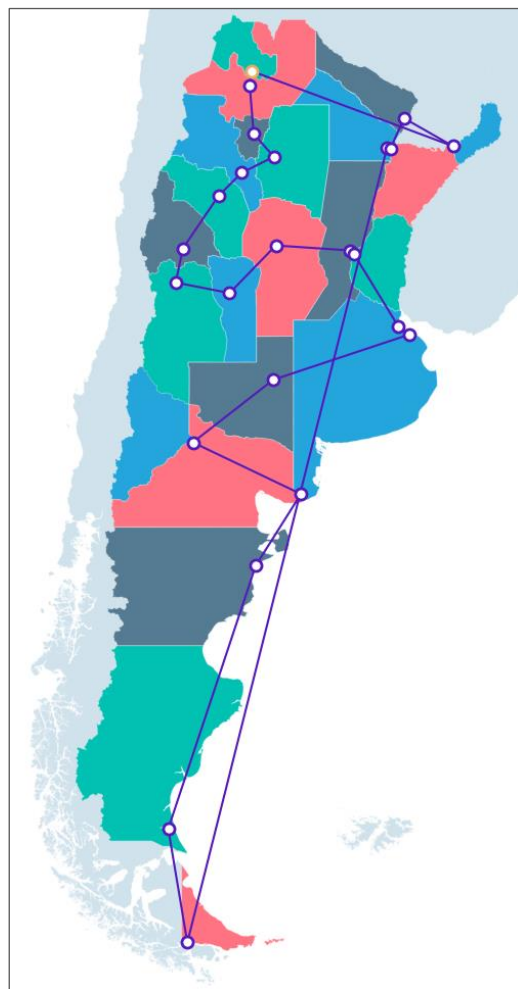
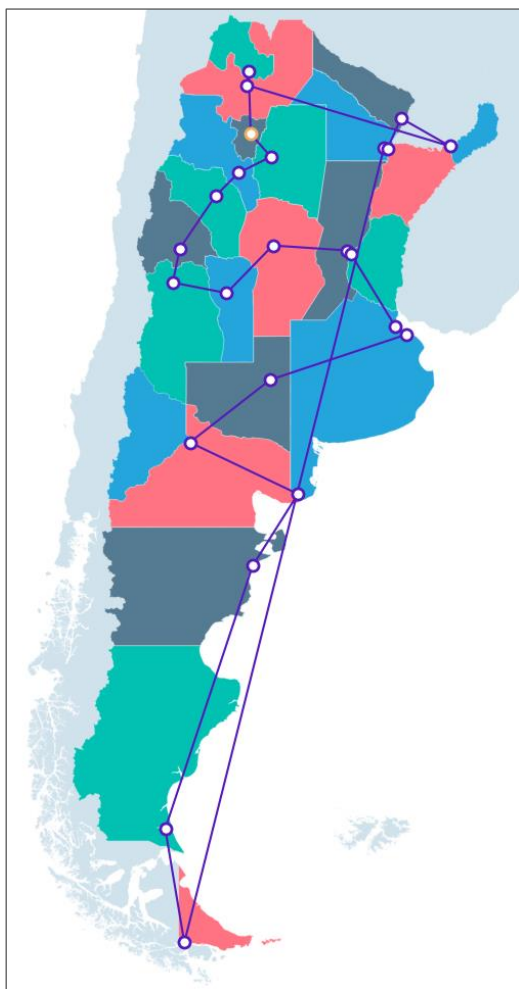


```
ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
14, San Miguel de Tucumán - 0 (0)
21, Santiago del Estero - 141 (141)
13, San Fernando del Valle de Catamarca - 166 (307)
5, La Rioja - 149 (456)
17, San Juan - 283 (739)
6, Mendoza - 152 (891)
18, San Luis - 235 (1126)
1, Córdoba - 293 (1419)
19, Santa Fe - 330 (1749)
8, Paraná - 19 (1768)
0, Ciudad Autónoma de Buenos Aires - 375 (2143)
4, La Plata - 53 (2196)
20, Santa Rosa - 602 (2798)
7, Neuquén - 422 (3220)
23, Viedma - 479 (3699)
10, Rawson - 327 (4026)
12, Río Gallegos - 975 (5001)
22, Ushuaia - 359 (5360)
11, Resistencia - 3130 (8490)
2, Corrientes - 13 (8503)
3, Formosa - 157 (8660)
9, Posadas - 263 (8923)
16, Salta - 992 (9915)
15, San Salvador de Jujuy - 67 (9982)
14, San Miguel de Tucumán - 293 (10275)

Distancia total recorrida: 10275
```

```
ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
15, San Salvador de Jujuy - 0 (0)
16, Salta - 67 (67)
14, San Miguel de Tucumán - 228 (295)
21, Santiago del Estero - 141 (436)
13, San Fernando del Valle de Catamarca - 166 (602)
5, La Rioja - 149 (751)
17, San Juan - 283 (1034)
6, Mendoza - 152 (1186)
18, San Luis - 235 (1421)
1, Córdoba - 293 (1714)
19, Santa Fe - 330 (2044)
8, Paraná - 19 (2063)
0, Ciudad Autónoma de Buenos Aires - 375 (2438)
4, La Plata - 53 (2491)
20, Santa Rosa - 602 (3093)
7, Neuquén - 422 (3515)
23, Viedma - 479 (3994)
10, Rawson - 327 (4321)
12, Río Gallegos - 975 (5296)
22, Ushuaia - 359 (5655)
11, Resistencia - 3130 (8785)
2, Corrientes - 13 (8798)
3, Formosa - 157 (8955)
9, Posadas - 263 (9218)
15, San Salvador de Jujuy - 1007 (10225)

Distancia total recorrida: 10225
```



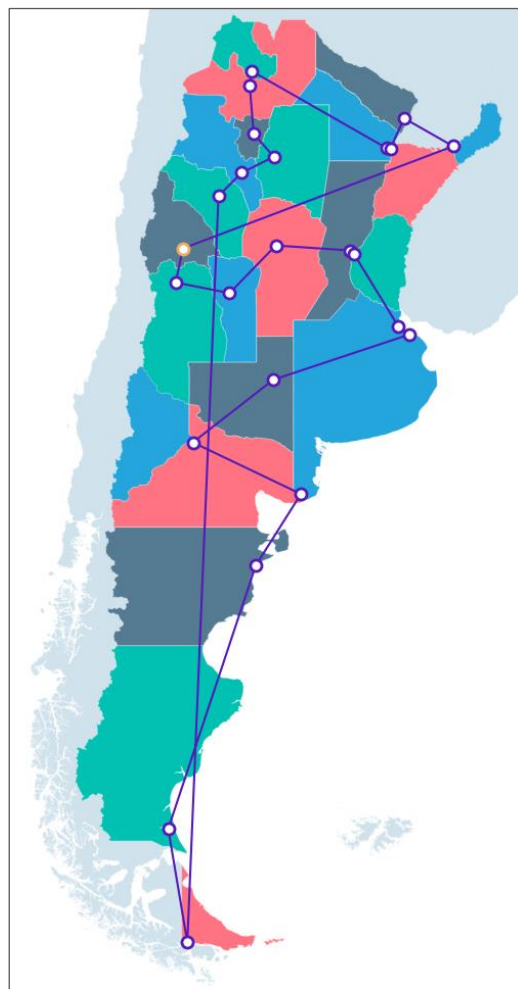
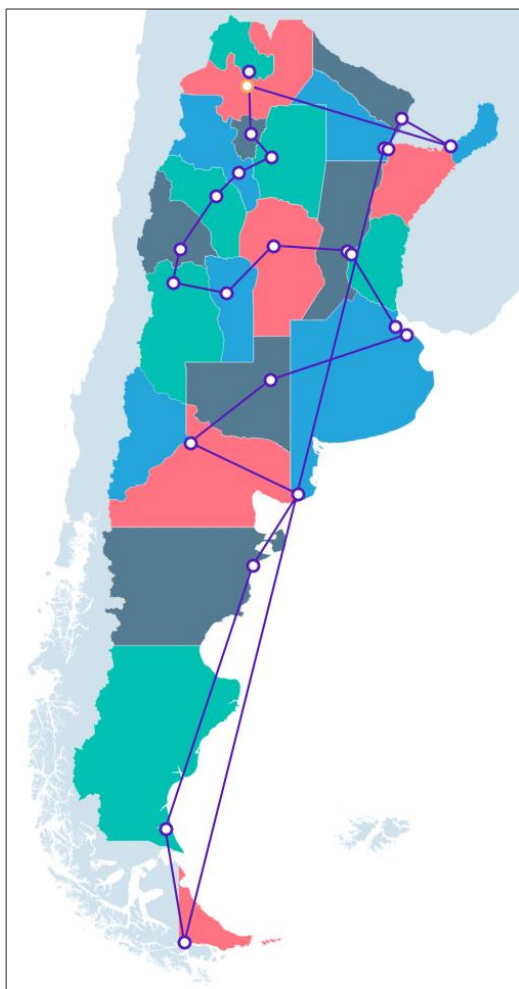


```
ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
16, Salta - 0 (0)
15, San Salvador de Jujuy - 67 (67)
14, San Miguel de Tucumán - 293 (360)
21, Santiago del Estero - 141 (501)
13, San Fernando del Valle de Catamarca - 166 (667)
5, La Rioja - 149 (816)
17, San Juan - 283 (1099)
6, Mendoza - 152 (1251)
18, San Luis - 235 (1486)
1, Córdoba - 293 (1779)
19, Santa Fe - 330 (2109)
8, Paraná - 19 (2128)
0, Ciudad Autónoma de Buenos Aires - 375 (2503)
4, La Plata - 53 (2556)
20, Santa Rosa - 602 (3158)
7, Neuquén - 422 (3580)
23, Viedma - 479 (4059)
10, Rawson - 327 (4386)
12, Río Gallegos - 975 (5361)
22, Ushuaia - 359 (5720)
11, Resistencia - 3130 (8850)
2, Corrientes - 13 (8863)
3, Formosa - 157 (9020)
9, Posadas - 263 (9283)
16, Salta - 992 (10275)

Distancia total recorrida: 10275
```

```
ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
17, San Juan - 0 (0)
6, Mendoza - 152 (152)
18, San Luis - 235 (387)
1, Córdoba - 293 (680)
19, Santa Fe - 330 (1010)
8, Paraná - 19 (1029)
0, Ciudad Autónoma de Buenos Aires - 375 (1404)
4, La Plata - 53 (1457)
20, Santa Rosa - 602 (2059)
7, Neuquén - 422 (2481)
23, Viedma - 479 (2960)
10, Rawson - 327 (3287)
12, Río Gallegos - 975 (4262)
22, Ushuaia - 359 (4621)
5, La Rioja - 2821 (7442)
13, San Fernando del Valle de Catamarca - 149 (7591)
21, Santiago del Estero - 166 (7757)
14, San Miguel de Tucumán - 141 (7898)
16, Salta - 228 (8126)
15, San Salvador de Jujuy - 67 (8193)
11, Resistencia - 729 (8922)
2, Corrientes - 13 (8935)
3, Formosa - 157 (9092)
9, Posadas - 263 (9355)
17, San Juan - 1306 (10661)

Distancia total recorrida: 10661
```





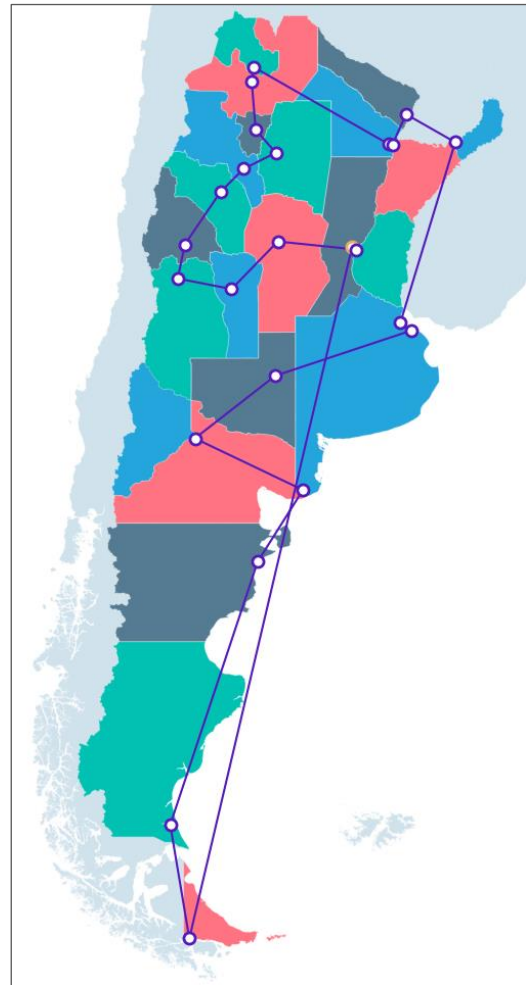
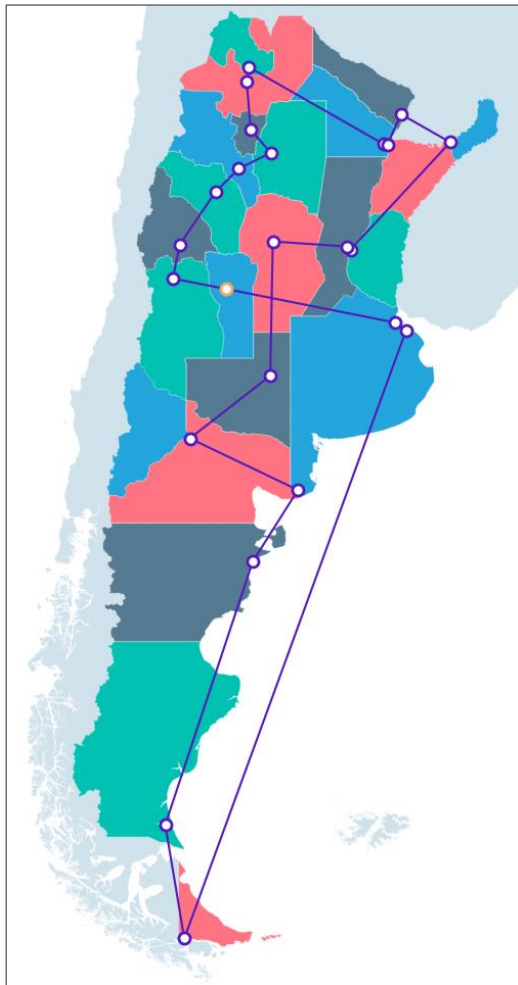


```
ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
18, San Luis - 0 (0)
6, Mendoza - 235 (235)
17, San Juan - 152 (387)
5, La Rioja - 283 (670)
13, San Fernando del Valle de Catamarca - 149 (819)
21, Santiago del Estero - 166 (985)
14, San Miguel de Tucumán - 141 (1126)
16, Salta - 228 (1354)
15, San Salvador de Jujuy - 67 (1421)
11, Resistencia - 729 (2150)
2, Corrientes - 13 (2163)
3, Formosa - 157 (2320)
9, Posadas - 263 (2583)
8, Paraná - 658 (3241)
19, Santa Fe - 19 (3260)
1, Córdoba - 330 (3590)
20, Santa Rosa - 577 (4167)
7, Neuquén - 422 (4589)
23, Viedma - 479 (5068)
10, Rawson - 327 (5395)
12, Río Gallegos - 975 (6370)
22, Ushuaia - 359 (6729)
4, La Plata - 2350 (9079)
0, Ciudad Autónoma de Buenos Aires - 53 (9132)
18, San Luis - 749 (9881)

Distancia total recorrida: 9881
```

```
ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
19, Santa Fe - 0 (0)
8, Paraná - 19 (19)
1, Córdoba - 348 (367)
18, San Luis - 293 (660)
6, Mendoza - 235 (895)
17, San Juan - 152 (1047)
5, La Rioja - 283 (1330)
13, San Fernando del Valle de Catamarca - 149 (1479)
21, Santiago del Estero - 166 (1645)
14, San Miguel de Tucumán - 141 (1786)
16, Salta - 228 (2014)
15, San Salvador de Jujuy - 67 (2081)
11, Resistencia - 729 (2810)
2, Corrientes - 13 (2823)
3, Formosa - 157 (2980)
9, Posadas - 263 (3243)
0, Ciudad Autónoma de Buenos Aires - 834 (4077)
4, La Plata - 53 (4130)
20, Santa Rosa - 602 (4732)
7, Neuquén - 422 (5154)
23, Viedma - 479 (5633)
10, Rawson - 327 (5960)
12, Río Gallegos - 975 (6935)
22, Ushuaia - 359 (7294)
19, Santa Fe - 2641 (9935)

Distancia total recorrida: 9935
```



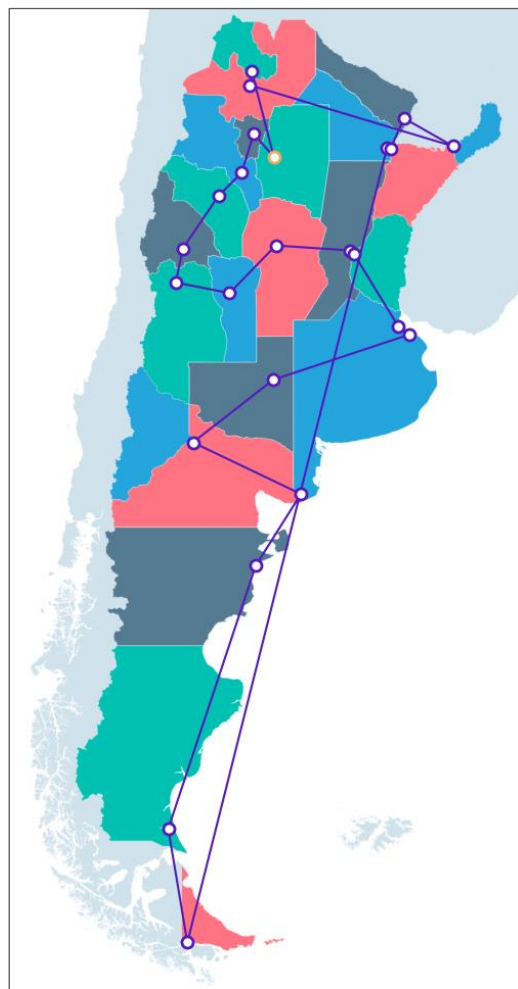
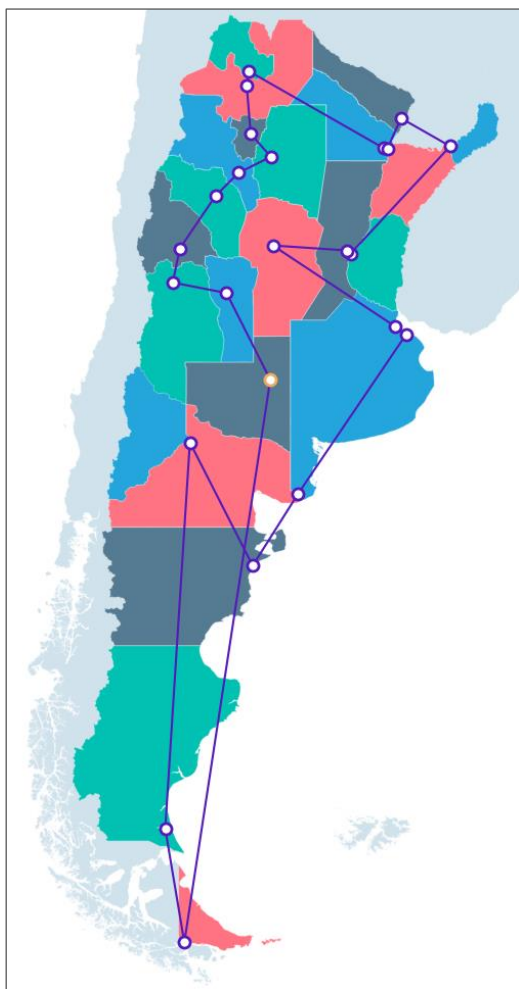


```
ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
20, Santa Rosa - 0 (0)
18, San Luis - 412 (412)
6, Mendoza - 235 (647)
17, San Juan - 152 (799)
5, La Rioja - 283 (1082)
13, San Fernando del Valle de Catamarca - 149 (1231)
21, Santiago del Estero - 166 (1397)
14, San Miguel de Tucumán - 141 (1538)
16, Salta - 228 (1766)
15, San Salvador de Jujuy - 67 (1833)
11, Resistencia - 729 (2562)
2, Corrientes - 13 (2575)
3, Formosa - 157 (2732)
9, Posadas - 263 (2995)
8, Paraná - 658 (3653)
19, Santa Fe - 19 (3672)
1, Córdoba - 330 (4002)
0, Ciudad Autónoma de Buenos Aires - 646 (4648)
4, La Plata - 53 (4701)
23, Viedma - 789 (5490)
10, Rawson - 327 (5817)
7, Neuquén - 543 (6360)
12, Río Gallegos - 1410 (7770)
22, Ushuaia - 359 (8129)
20, Santa Rosa - 2044 (10173)

Distancia total recorrida: 10173
```

```
ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
21, Santiago del Estero - 0 (0)
14, San Miguel de Tucumán - 141 (141)
13, San Fernando del Valle de Catamarca - 189 (330)
5, La Rioja - 149 (479)
17, San Juan - 283 (762)
6, Mendoza - 152 (914)
18, San Luis - 235 (1149)
1, Córdoba - 293 (1442)
19, Santa Fe - 330 (1772)
8, Paraná - 19 (1791)
0, Ciudad Autónoma de Buenos Aires - 375 (2166)
4, La Plata - 53 (2219)
20, Santa Rosa - 602 (2821)
7, Neuquén - 422 (3243)
23, Viedma - 479 (3722)
10, Rawson - 327 (4049)
12, Río Gallegos - 975 (5024)
22, Ushuaia - 359 (5383)
11, Resistencia - 3130 (8513)
2, Corrientes - 13 (8526)
3, Formosa - 157 (8683)
9, Posadas - 263 (8946)
16, Salta - 992 (9938)
15, San Salvador de Jujuy - 67 (10005)
21, Santiago del Estero - 414 (10419)

Distancia total recorrida: 10419
```





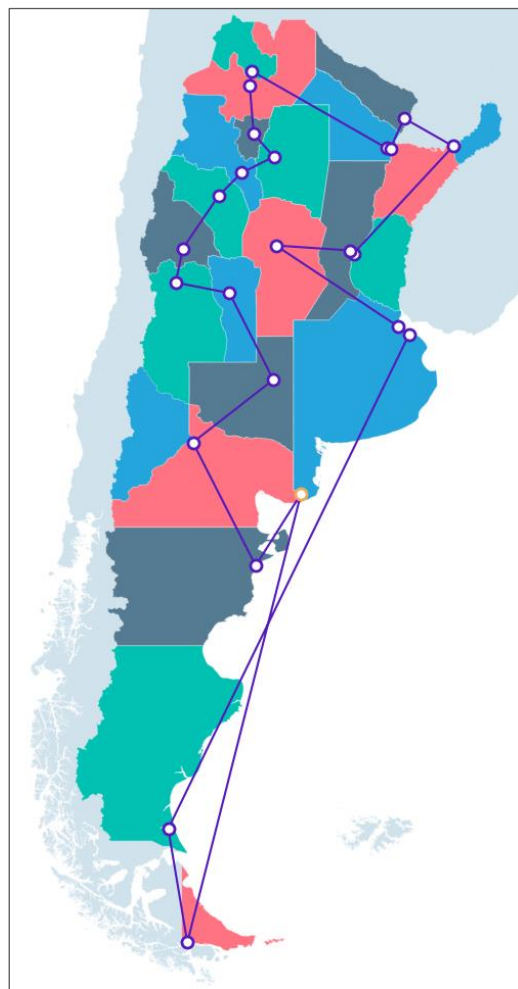
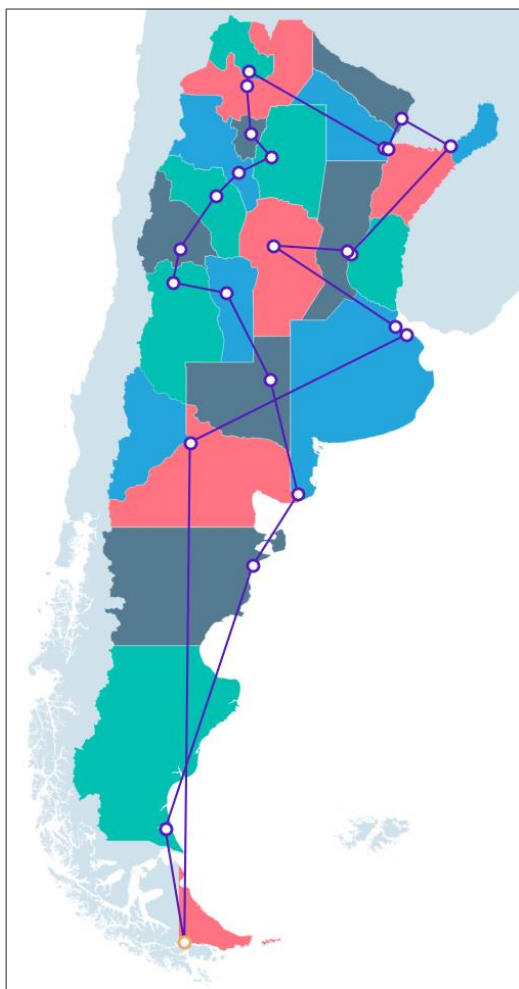


```
ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
22, Ushuaia - 0 (0)
12, Rio Gallegos - 359 (359)
10, Rawson - 975 (1334)
23, Viedma - 327 (1661)
20, Santa Rosa - 477 (2138)
18, San Luis - 412 (2550)
6, Mendoza - 235 (2785)
17, San Juan - 152 (2937)
5, La Rioja - 283 (3220)
13, San Fernando del Valle de Catamarca - 149 (3369)
21, Santiago del Estero - 166 (3535)
14, San Miguel de Tucumán - 141 (3676)
16, Salta - 228 (3904)
15, San Salvador de Jujuy - 67 (3971)
11, Resistencia - 729 (4700)
2, Corrientes - 13 (4713)
3, Formosa - 157 (4870)
9, Posadas - 263 (5133)
8, Paraná - 658 (5791)
19, Santa Fe - 19 (5810)
1, Córdoba - 330 (6140)
0, Ciudad Autónoma de Buenos Aires - 646 (6786)
4, La Plata - 53 (6839)
7, Neuquén - 1005 (7844)
22, Ushuaia - 1762 (9606)

Distancia total recorrida: 9606
```

```
ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
23, Viedma - 0 (0)
10, Rawson - 327 (327)
7, Neuquén - 543 (870)
20, Santa Rosa - 422 (1292)
18, San Luis - 412 (1704)
6, Mendoza - 235 (1939)
17, San Juan - 152 (2091)
5, La Rioja - 283 (2374)
13, San Fernando del Valle de Catamarca - 149 (2523)
21, Santiago del Estero - 166 (2689)
14, San Miguel de Tucumán - 141 (2830)
16, Salta - 228 (3058)
15, San Salvador de Jujuy - 67 (3125)
11, Resistencia - 729 (3854)
2, Corrientes - 13 (3867)
3, Formosa - 157 (4024)
9, Posadas - 263 (4287)
8, Paraná - 658 (4945)
19, Santa Fe - 19 (4964)
1, Córdoba - 330 (5294)
0, Ciudad Autónoma de Buenos Aires - 646 (5940)
4, La Plata - 53 (5993)
12, Rio Gallegos - 2064 (8057)
22, Ushuaia - 359 (8416)
23, Viedma - 1605 (10021)

Distancia total recorrida: 10021
```





A modo de resumen, la siguiente tabla enumera los resultados de las búsquedas heurísticas para cada ciudad de partida.

Ciudad de partida	Longitud del recorrido (km)
Ciudad Autónoma de Buenos Aires	9933
Córdoba	9781
Corrientes	10840
Formosa	11031
La Plata	9858
La Rioja	10237
Mendoza	10010
Neuquén	9335
Paraná	9911
Posadas	10767
Rawson	9579
Resistencia	10835
Río Gallegos	9579
San Fernando del Valle de Catamarca	10647
San Miguel de Tucumán	10275
San Salvador de Jujuy	10225
Salta	10275
San Juan	10661
San Luis	9881
Santa Fe	9935
Santa Rosa	10173
Santiago del Estero	10419
Ushuaia	9606
Viedma	10021



## Resultados 2 - B (Mínimo por Búsqueda Heurística)

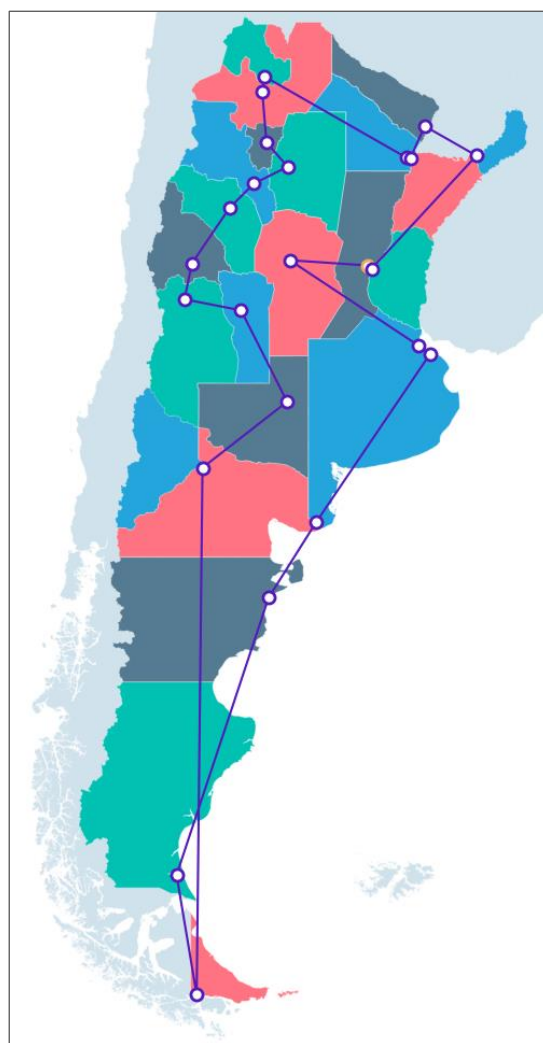
Al seleccionar esta opción, el programa realizará una búsqueda heurística por cada ciudad de partida y tomará como resultado final al recorrido de menor longitud. Luego, “rotará” este recorrido hasta que coincida con la ciudad de partida elegida.

El recorrido de menor longitud siempre será el resultado de la búsqueda heurística con Neuquén como ciudad de partida, ya que como se puede ver en la tabla de la página anterior, la longitud de dicho recorrido (9335 *km*), es la más pequeña.

Así se ven dos resultados con ciudades de partidas diferentes:

ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
19, Santa Fe - 0 (0)
1, Córdoba - 330 (330)
0, Ciudad Autónoma de Buenos Aires - 646 (976)
4, La Plata - 53 (1029)
23, Viedma - 789 (1818)
10, Rawson - 327 (2145)
12, Río Gallegos - 975 (3120)
22, Ushuaia - 359 (3479)
7, Neuquén - 1762 (5241)
20, Santa Rosa - 422 (5663)
18, San Luis - 412 (6075)
6, Mendoza - 235 (6310)
17, San Juan - 152 (6462)
5, La Rioja - 283 (6745)
13, San Fernando del Valle de Catamarca - 149 (6894)
21, Santiago del Estero - 166 (7060)
14, San Miguel de Tucumán - 141 (7201)
16, Salta - 228 (7429)
15, San Salvador de Jujuy - 67 (7496)
11, Resistencia - 729 (8225)
2, Corrientes - 13 (8238)
3, Formosa - 157 (8395)
9, Posadas - 263 (8658)
8, Paraná - 658 (9316)
19, Santa Fe - 19 (9335)
Distancia total recorrida: 9335

ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
12, Río Gallegos - 0 (0)
22, Ushuaia - 359 (359)
7, Neuquén - 1762 (2121)
20, Santa Rosa - 422 (2543)
18, San Luis - 412 (2955)
6, Mendoza - 235 (3190)
17, San Juan - 152 (3342)
5, La Rioja - 283 (3625)
13, San Fernando del Valle de Catamarca - 149 (3774)
21, Santiago del Estero - 166 (3940)
14, San Miguel de Tucumán - 141 (4081)
16, Salta - 228 (4309)
15, San Salvador de Jujuy - 67 (4376)
11, Resistencia - 729 (5105)
2, Corrientes - 13 (5118)
3, Formosa - 157 (5275)
9, Posadas - 263 (5538)
8, Paraná - 658 (6196)
19, Santa Fe - 19 (6215)
1, Córdoba - 330 (6545)
0, Ciudad Autónoma de Buenos Aires - 646 (7191)
4, La Plata - 53 (7244)
23, Viedma - 789 (8033)
10, Rawson - 327 (8360)
12, Río Gallegos - 975 (9335)
Distancia total recorrida: 9335

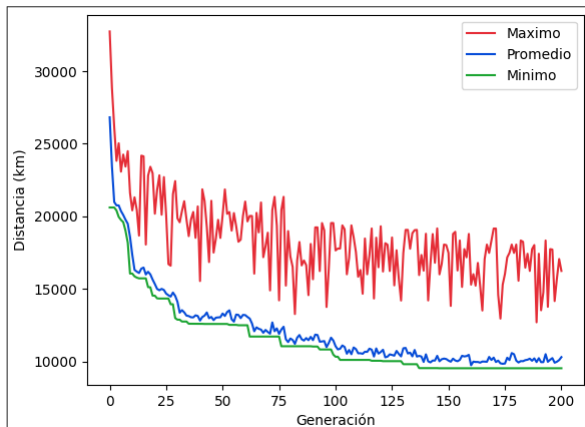


El mapa que se encuentra a la derecha corresponde al recorrido iniciado en la ciudad de Santa Fe. Sin embargo, el recorrido de cualquier otra ejecución con una ciudad de partida diferente generaría el mismo recorrido.

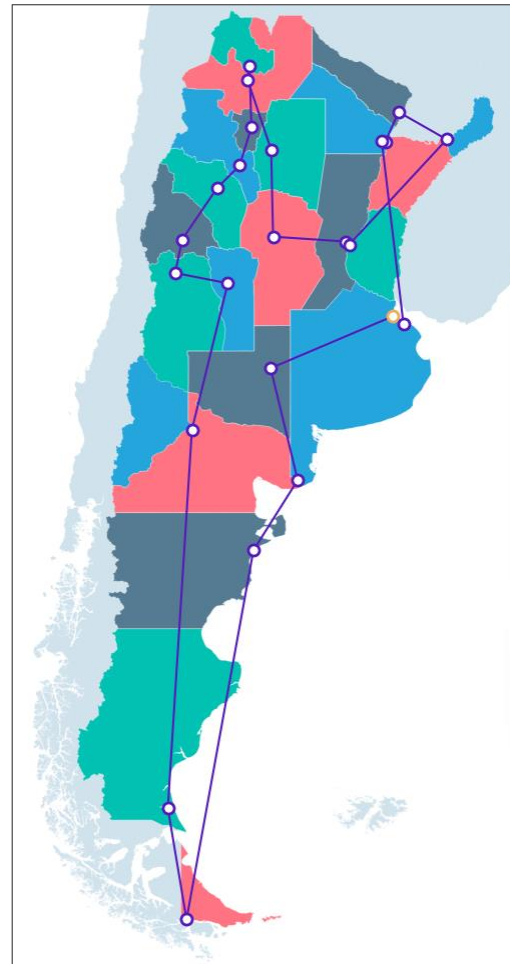


## Resultados 2 - C (Algoritmo Genético)

A continuación, se encuentra un resultado obtenido usando Algoritmos Genéticos.



```
ID, CIUDAD - DISTANCIA DEL SEGMENTO (DISTANCIA ACUMULADA)
0, Ciudad Autónoma de Buenos Aires - 0 (0)
20, Santa Rosa - 579 (579)
23, Viedma - 477 (1056)
10, Rawson - 327 (1383)
22, Ushuaia - 1300 (2683)
12, Río Gallegos - 359 (3042)
7, Neuquén - 1410 (4452)
18, San Luis - 643 (5095)
6, Mendoza - 235 (5330)
17, San Juan - 152 (5482)
5, La Rioja - 283 (5765)
13, San Fernando del Valle de Catamarca - 149 (5914)
14, San Miguel de Tucumán - 189 (6103)
15, San Salvador de Jujuy - 293 (6396)
16, Salta - 67 (6463)
21, Santiago del Estero - 353 (6816)
1, Córdoba - 401 (7217)
19, Santa Fe - 330 (7547)
8, Paraná - 19 (7566)
9, Posadas - 658 (8224)
3, Formosa - 263 (8487)
2, Corrientes - 157 (8644)
11, Resistencia - 13 (8657)
4, La Plata - 833 (9490)
0, Ciudad Autónoma de Buenos Aires - 53 (9543)
Distancia total recorrida: 9543
```



En general, las ejecuciones del algoritmo generaban como salida recorridos con longitudes entre 9500 km y 13500 km. Asumiendo que el promedio se encuentra justo en la mitad de dicho intervalo, este sería 11500 km. Esto quiere decir que la salida esperada del algoritmo genético se encuentra generalmente por encima de los resultados obtenidos a través de la búsqueda heurística, aunque en algunos casos puede generar resultados mejores.

En particular, el ejemplo mostrado arriba, donde la longitud del recorrido es de 9534 km, es mejor que 23 de los 24 resultados heurísticos (95,83%), siendo superado solo por el recorrido heurístico partiendo desde Neuquén (9335 km).

Al realizar más ejecuciones y variaciones en los parámetros iniciales, sería posible eventualmente conseguir resultados mejores a los heurísticos.



## Conclusiones

Como podemos ver en algunos problemas como el TSP, no siempre es posible explorar el espacio solución completo para hallar la solución óptima al problema. Sin embargo, en este tipo de problemas, siempre existen métodos para generar soluciones que, sean mejores o peores, pueden ser satisfactorias a nuestro problema

Cuando se tiene la suficiente información para definir una función heurística, los métodos heurísticos son muy útiles para encontrar soluciones aceptables en muy poco tiempo.

Por otro lado, también se pudo ver que los algoritmos genéticos son otra alternativa muy buena a la falta de poder realizar búsquedas exhaustivas, generando soluciones óptimas y pudiendo llegar a superar a las búsquedas heurísticas en algunos casos, con el costo de utilizar más recursos computacionales y algo más de tiempo.