



**Diseño de rutas para el reparto de vacunas COVID-19 en la red hospitalaria de Antioquía utilizando Búsqueda Tabú**

**Presentado por:**

**Kevin David Hernández Molina**

**Tutor:**

**Albeiro Hernán Suarez Hernández**

**Cotutor:**

**Gustavo Andrés Araque González**

Institución Universitaria Politécnico Grancolombiano  
Ingeniería Industrial  
Medellín, Colombia  
2021

1.	RESUMEN DEL PROYECTO:	7
2.	TITULO DEL PROYECTO:	7
3.	PLANTEAMIENTO DEL PROBLEMA:	7
4.	OBJETIVO GENERAL	9
5.	OBJETIVOS ESPECIFICOS	9
6.	JUSTIFICACIÓN:	10
7.	MARCO TEÓRICO	11
8.	METODOLOGÍA	16
8.1.	Descripción cuantitativa del problema	16
8.2.	Definición de las variables de decisión modelo 1 (matriz de distancias):	16
8.3.	Definición de las variables de decisión modelo 2 (matriz de tiempos):	16
8.4.	Definición de la función objetivo modelo 2	17
8.5.	Definición de la función objetivo modelo 1	17
8.6.	Restricciones del modelo:	17
8.7.	Definición de variables relevantes y supuestos	17
8.8.	Variables relevantes	18
8.9.	Selección y programación del método de resolución (Metaheurística)	19
8.10.	Explicación de la metodología empleada en la programación:	23
8.11.	Pasos con ejemplo tomados de (S.Hillier & J. Lieberman, 2010):	23
8.12.	Pasos de la metodología búsqueda tabú con sub-viaje inverso en macro Excel:	26
8.13.	Método de mutación adaptada	35
8.14.	Evitar repetidos en la mutación:	40
8.15.	Caso con nodo ocupado en la mutación:	42
9.	HIPOTESIS	44
10.	RESULTADOS:	45
11.	ANÁLISIS DE COSTOS:	46
12.	MEDICIÓN DE LA CALIDAD:	46
13.	RECOMENDACIONES DEL ALGORITMO DE MUTACION:	47
14.	CONCLUSIONES	48
15.	DISCUSIÓN Y TRABAJOS FUTUROS	49
16.	CUMPLIMIENTO DE LOS OBJETIVOS DEL PROYECTO	50

17.	CRONOGRAMA .....	51
18.	BIBLIOGRAFÍA.....	52
19.	ANEXOS:.....	54

## Lista de figuras

	Pág.
<b>Figura 1 :</b> .....	<b>16</b>
<b>Figura 2 :</b> .....	<b>21</b>
<b>Figura 3 :</b> .....	<b>22</b>
<b>Figura 4 :</b> .....	<b>22</b>
<b>Figura 5 :</b> .....	<b>26</b>
<b>Figura 6 :</b> .....	<b>27</b>
<b>Figura 7 :</b> .....	<b>27</b>
<b>Figura 8 :</b> .....	<b>27</b>
<b>Figura 9 :</b> .....	<b>28</b>
<b>Figura 10 :</b> .....	<b>28</b>
<b>Figura 11 :</b> .....	<b>29</b>
<b>Figura 12 :</b> .....	<b>29</b>
<b>Figura 13 :</b> .....	<b>30</b>
<b>Figura 14 :</b> .....	<b>30</b>
<b>Figura 15 :</b> .....	<b>31</b>
<b>Figura 16 :</b> .....	<b>31</b>
<b>Figura 17 :</b> .....	<b>32</b>
<b>Figura 18 :</b> .....	<b>32</b>
<b>Figura 19 :</b> .....	<b>33</b>
<b>Figura 20 :</b> .....	<b>34</b>
<b>Figura 21 :</b> .....	<b>35</b>
<b>Figura 22 :</b> .....	<b>36</b>
<b>Figura 23 :</b> .....	<b>36</b>
<b>Figura 24 :</b> .....	<b>37</b>
<b>Figura 25 :</b> .....	<b>38</b>
<b>Figura 26 :</b> .....	<b>39</b>
<b>Figura 27 :</b> .....	<b>40</b>
<b>Figura 28 :</b> .....	<b>41</b>
<b>Figura 29 :</b> .....	<b>41</b>
<b>Figura 30 :</b> .....	<b>42</b>
<b>Figura 31 :</b> .....	<b>42</b>
<b>Figura 32 :</b> .....	<b>42</b>
<b>Figura 33 :</b> .....	<b>43</b>

<b>Figura 34 :</b> .....	<b>43</b>
<b>Figura 35 :</b> .....	<b>44</b>

## Lista de tablas

	Pág.
<b>Tabla 1 :</b> .....	<b>15</b>
<b>Tabla 2 :</b> .....	<b>19</b>
<b>Tabla 3 :</b> .....	<b>20</b>
<b>Tabla 4 :</b> .....	<b>20</b>
<b>Tabla 5 :</b> .....	<b>21</b>
<b>Tabla 6 :</b> .....	<b>23</b>
<b>Tabla 7 :</b> .....	<b>24</b>
<b>Tabla 8 :</b> .....	<b>45</b>
<b>Tabla 9 :</b> .....	<b>46</b>
<b>Tabla 10 :</b> .....	<b>46</b>
<b>Tabla 11 :</b> .....	<b>46</b>
<b>Tabla 12 :</b> .....	<b>50</b>
<b>Tabla 13 :</b> .....	<b>51</b>

## **1. RESUMEN DEL PROYECTO:**

Gracias a la llegada de las vacunas contra el COVID-19 al país, el gobierno nacional en cabeza del Ministerio de Salud y Protección social enfrenta el duro proceso de distribuir los biológicos a las diferentes bodegas gubernamentales sin alteraciones en su cadena de frío, con el fin de hacerlas llegar a los diferentes centros hospitalarios y mitigar las consecuencias ya desastrosas que ha generado la pandemia del coronavirus. Para lograr este objetivo la presente investigación se centra en el desarrollo de rutas de reparto de vacunas por medio de 2 modelos distintos (Modelo 1 minimización de distancia total de ruta y Modelo 2 minimización de tiempo de recorrido de la ruta), que propone posibles rutas logísticas de reparto de los inmunológicos contra el COVID-19 en una muestra adaptable de 40 hospitales del valle de aburra categorizándolo como un problema de Agente viajero de complejidad de resolución: NP-completo, uno de los problemas más difíciles de la optimización combinatoria, debido al alto número de variables a tratar (1681 variables) y al número de posibles rutas factibles ( $8,16 * 10^{47}$ ). Este proyecto parte del desarrollo del proyecto de investigación formativa del curso de metaheurísticas, con el que se decide usar una metaheurística llamada búsqueda tabú, y se adiciona un nuevo operador de búsqueda basado en una mutación genética que deriva del uso del gas mostaza en la primera guerra mundial a fin de desarrollar una metodología de adaptación que combina el uso de la heurística el vecino más cercano con el algoritmo clásico de Búsqueda Tabú.

Palabras Claves: Mutación, Metaheurística, Agente viajero, Búsqueda tabú, Logística de distribución.

## **2. TITULO DEL PROYECTO:**

Diseño de rutas para el reparto de vacunas COVID-19 en la red hospitalaria de Antioquía utilizando Búsqueda Tabú

## **3. PLANTEAMIENTO DEL PROBLEMA:**

La extensión del territorio nacional alberga una amplia variedad de bosques y ecosistemas que requieren atravesar múltiples carreteras para lograr desplazar diversas cargas o personas a través del país. Dependiendo del estado actual de la red vial, se puede determinar la distancia y el tiempo de múltiples rutas, que permitan llegar más rápido o no, hacia su destino final. La problemática radica en que existen diferentes secuencias para realizar el recorrido y múltiples vías que llegan al mismo lugar, de acuerdo con las circunstancias, es necesario el transporte terrestre rápido y efectivo de las vacunas que deben mantener la cadena de frío. Para lograr esto se emplea la optimización combinatoria de un problema clásico del agente viajero (TSP), de tamaño  $n$  (número de puntos a visitar), definida por el número de restricciones ( $2*n$ ), número variables ( $n*n$ ) y número de recorridos posibles  $(n-1)!$ , que,

acorde con Lopez, mendoza v., & cuartas, (2008) “su dificultad está ligada al tiempo de ejecución del ejercicio, es decir, el tiempo que le toma a un ordenador en llegar a una solución óptima”, así como también el número de “n” a tratar, ya que de acuerdo con Cruz Chávez et al., (2014), “cuando el número de variables que componen el problema es una cantidad extremadamente grande, el problema se vuelve cada vez más difícil de resolver en un tiempo razonable” ; lo que lo clasifica, según Bellidolo, (2013), como uno de los problemas de optimización combinatoria más completos (NP-duro).

Actualmente existen en el mercado múltiples herramientas computacionales que permiten resolver este tipo de problema; y algunos de ellos son: GAMS (sistema general de modelado algebraico), GUROBI, CPEX, Lingo y el Complemento Solver de Excel. Para el caso del software CPLEX, de acuerdo con (*Nodo Optimización de CPLEX - IBM Documentation, s/f*), puede procesar hasta un máximo de 1000 variables, volviéndolo ineficiente para esta investigación, puesto que los modelos 1 y 2 a tratar poseen exactamente 1681. Para el caso del software Lingo, según (Ivorra, s/f), este posee una limitación parecida a la del complemento Solver de Excel que restringe el cálculo de problemas debido a la complejidad de estos, haciendo que el programa arroje error durante su ejecución. En el caso del complemento Solver de Excel, este utiliza el algoritmo Simplex, que ha ayudado a resolver este tipo de problema de manera exacta, pero, este programa tiene sus limitaciones, que, según Garrido Lopez et al., (2009), “solo permite hasta un máximo de 200 variables”, lo que ha conllevado a la búsqueda de alternativas de cálculo para una solución más aproximada de este tipo de dificultad. Para el caso de GAMS, este software es un poco más robusto en términos de procesamiento, que el complemento Solver de Excel, pero, al igual que este, solo permite procesar 300 variables de las 1681 que poseen los modelos 1 y 2 que se abordan en esta problemática. Para el caso de GUROBI, este software, permite resolver el problema de manera exacta mediante sus propios servidores en Estados Unidos, pero, aunque logra solucionar esta problemática de forma exacta, el tiempo en que tarda en dar solución puede tardar varios días, lo que en términos de tiempo de procesamiento es muy demorado a la hora de adaptarlo hacia nuevos proyectos que exijan el resultado en una ventana de tiempo muy corto, la decisión de secuenciación de puntos de reparto es una decisión de corto plazo y es necesario aplicar una metodología que en corto tiempo suministre una ruta satisfactoria en cuanto a la distancia recorrida o el tiempo total del reparto de vacunas.

Debido a la complejidad y la necesidad de transportar las vacunas por tierra, utilizando rutas con menor distancias o con menor tiempo de recorrido, se pretende diseñar diferentes rutas de reparto para las vacunas, mediante un algoritmo basado en una mutación existente derivada de la primera guerra mundial, que, combina el uso de la heurística “el vecino más cercano” y la metaheurística búsqueda tabú, para dar solución a esta problemática.

De lo anterior se puede inferir la siguiente pregunta a resolver:

¿Cómo diseñar un sistema de rutas utilizando un operador de mutación del algoritmo de búsqueda tabú clásico, que permita mejorar las trayectorias en cuanto al tiempo y distancias para el transporte y distribución de las vacunas COVID-19 en una muestra de la red hospitalaria del valle de aburra?

#### **4. OBJETIVO GENERAL**

Diseñar rutas de reparto de vacunas COVID-19 desde el almacén distrital a la red hospitalaria de Antioquia, utilizando una mejora en la estrategia del algoritmo clásico de Búsqueda Tabú.

#### **5. OBJETIVOS ESPECIFICOS**

- Realizar una revisión bibliográfica de las diferentes estrategias que se han utilizado para resolver problemas de ruteo, haciendo énfasis en aquellos artículos que utilizaron como estrategia central un algoritmo de búsqueda tabú.
- Modelar matemáticamente el problema de ruteo y recopilar información de los parámetros relacionados con distancia y tiempo entre los diferentes hospitales de la Red de Antioquia.
- Ejecutar el algoritmo de Búsqueda Tabú clásico y agregar un nuevo operador que mejore la estrategia de búsqueda.
- Determinar las rutas que mejoren la distancia y el tiempo total de entrega de vacunas COVID19 en una muestra de la red hospitalaria de Antioquia.

## 6. JUSTIFICACIÓN:

Debido a la cadena de frío a las que deben estar sometidas las vacunas para conservarse que, de acuerdo con el respectivo fabricante, estas deben de estar a  $-70^{\circ}\text{C}$  (para el caso de Pfizer y BioNTech),  $-20^{\circ}\text{C}$  (para el caso de Moderna), entre  $-25^{\circ}\text{C}$  a  $-15^{\circ}\text{C}$  (para el caso Oxford/AstraZeneca y Sinovac) y  $-18^{\circ}\text{C}$  (para el caso de Sputnik V), deben ser trasladadas mediante camiones refrigerantes especiales cuya temperatura varia alrededor de los  $-78.3^{\circ}\text{C}$ , trabajando a plena potencia sin tener en cuenta el gasto del combustible, así como el estado de las vías durante el trayecto hacia las entidades de salud de destino. Para llevar a cabo esta logística es necesaria la planeación de las rutas de distribución que permitan minimizar la distancia recorrida y/o el tiempo de la ruta entre el almacén distrital hasta su destino, ya que se corre el riesgo de romper la cadena de frío durante el transporte y por ende su poder inmunológico.

Un claro ejemplo de lo delicado que puede llegar a ser el transporte de estas vacunas se vivió el 26 de febrero de 2021, en la ciudad de Tolima, Colombia, que según, Torres Cepeda, (2021) en donde, por errores en la logística del transporte, el ministerio de salud se vio en la obligación de colocar en cuarentena más de 5700 vacunas, gracias a que, dichos biológicos presentaron una alteración en la cadena de frío, con una alta posibilidad de pérdida de su poder inmunológico, lo que conllevaría a pérdidas económicas, de tiempo y esfuerzo por parte del gobierno nacional que desde el comienzo de la pandemia a tratado de mitigar y proteger a la población.

El uso de metaheurísticas en lugar de algoritmos exactos para resolver problemas de optimización combinatoria se justifica cuando, sabiendo que existe un método exacto para resolver el problema, su uso es computacionalmente muy costoso o inviable. La decisión de ruteo es de corto plazo y las herramientas computacionales actuales, como el caso de GUROBI, pueden tardar días de procesamiento para llegar a la ruta óptima.

Aunque el problema de investigación se aborda desde un problema clásico de Agente Viajero con tamaño  $n=41$ , se realizan algunos supuestos que permiten simplificar el modelamiento matemático y obtener en un tiempo de procesamiento corto rutas factibles con buen desempeño respecto a la distancia total recorrida o el tiempo de la trayectoria total.

## 7. MARCO TEÓRICO

### 7.1. Marco referencial

A continuación, se referencian algunas investigaciones relevantes para la realización de este proyecto:

En el año 2020 Morales Alvis , (Morales, Jose Luis, 2004) desarrolló, a una empresa en la ciudad de Bogotá, Colombia, dedicada transporte de valores, un estudio de productividad de recogida y entrega de dinero a cada cliente, dependiendo de sus necesidades y el lugar a donde deseaban que los transportaran, el resultado (medido en una escala de número de servicios prestados por hora), en comparación con otros países como México (con un 2.3 en servicio por hora), arrojó que la compañía poseía un servicio por hora de 1.4, lo que indicó un bajo desempeño, por lo tanto, este trabajo se centró en buscar e identificar el tipo de modelo de ruteo que permitiera optimizar esta operación logística, involucrando modelos de rutas que incorporen el uso de herramientas informáticas que no permitan depender de la experiencia de los conductores para elegir como llegar de un punto a otro; incluso plantear la forma de como involucrar los cambios normales de variables durante el día tales como tráfico, novedades de capacidad, ventanas de tiempo, cancelaciones y solicitudes de servicios especiales por parte de los clientes.

Para solucionar este problema de ruteo, se usó el algoritmo búsqueda tabú, que, si bien recordamos según Glover & Melian, (2003), es un procedimiento metaheurístico cuya característica distintiva es el uso de memoria adaptativa, previniendo el regreso a las rutas antiguas, logrando así una mejor solución al problema de la compañía.

En otro estudio realizado por Delgado Betancur et al., (2019), en la zona norte de la ciudad de Medellín, Colombia, en donde se evidenció una problemática del transporte público, debido a la alta densidad poblacional, que provocaba que el sistema fuera ineficiente en términos de tiempo (duración del viaje) y en frecuencia de autobuses.

Para dar solución a esta problemática, los investigadores utilizaron diferentes métodos metaheurísticos como lo es búsqueda tabú, mejorando significativamente la velocidad de los viajes que estaban alrededor de los 1800 segundos, mejorando a 1620 segundos, es decir que pasó de media hora (30 minutos) a 23 minutos, logrando un ahorro de 7 minutos por travesía, garantizando la cobertura del servicio público en las horas más extenuantes del día (horas pico).

Por otro lado, en la ciudad de México, Melendez Acosta, (2020), desarrollo una solución del agente viajero que combina las metaheurísticas de recocido simulado y algoritmo genético, que, este utilizó temperaturas de 20 a 50 con una velocidad de enfriamiento igualmente de 20 a 50 y un grado de descenso de temperatura de 0.97. Mientras que con algoritmo genético decidieron usar un factor de clonación del 0.5, de mutación del 0.1, con una población de 100 a 300. Utilizando combinación de ciudades al azar de entre 10 a 50, lograron establecer que

el mejor algoritmo de entre los dos mencionados fue recocido simulado, esto debido a la aleatoriedad usada entre las ciudades que redujo el número de iteraciones, pero el en algoritmo genético las aumentó para llegar a una respuesta óptima. Al no partir de una ruta factible desmejora la solución del algoritmo genético lo que no pasa con recocido simulado que depende de las temperaturas establecidas inicialmente.

De lo anterior y teniendo en cuenta el vasto territorio geográfico colombiano, se pretende diseñar posibles rutas logísticas que permita ahorra tiempo y distancia y evitando lo que según Torres Cepeda, (2021), pasó en Tolima, en donde debido a una mala logística en el transporte, supuestamente, se habrían alterado la cadena de frío de más de 5700 vacunas, las cuales fueron puestas en cuarentena para su análisis por parte de la farmacéutica Sinovac con fin de determinar si se podían usar o finalmente debían desecharse.

De acuerdo con el gobierno nacional, el Ministerio de Salud y protección social, (2021) pese a la pérdida de temperatura de las vacunas, el producto sigue siendo estable sin sufrir algún daño aparente de sus propiedades inmunológicas, ya que pueden tolerar temperaturas hasta de 25°C por 14 días.

Otro caso que se vivió el 13 de mayo de 2021 en la ciudad de Cundinamarca, Colombia, en donde según (*COVID-19: En Cundinamarca se perdieron más de 7.000 vacunas por interrumpir cadena de frío | Regiones | W Radio Colombia, s/f*) por errores humanos se perdieron más de 7000 vacunas de la farmacéutica Pfizer, demostrando lo delicado que puede ser la manipulación de los biológico y los susceptibles que son a altas temperaturas.

Con el fin de evitar lo ocurrido en Cundinamarca, Colombia, en donde se evidencio las consecuencias del rompimiento de la cadena de frío en los biológicos de Pfizer, se pretende diseñar diferentes rutas logísticas de reparto de vacunas COVID-19, dividido en dos diferentes modelos (modelo 1 de minimización de distancias y modelo 2 de minimización de tiempos), usando una nueva característica en el uso de la búsqueda tabú que parte de la biología, más explícitamente de una mutación biológica, que, de acuerdo con Klug et al., (2012), las define como una alteración de la secuencia de ADN, es decir una alteración en los procesos biológicos que si bien, en trabajos anteriores como el de Roberto et al., (2008), que utilizando ratas de mutación en combinación con el método Gris-Las, logro un porcentaje de error del 1% y en el peor de los casos del 1.9%, consiguiendo el mejor de los resultados en comparación con el modelo de islas del mismo informe. En tal sentido viendo que es posible aplicar mutaciones y no operadores de cruce, característicos de los algoritmos genéticos que, de acuerdo con Magallanes Bernal et al., (2021), la define como operaciones de optimización basados en la teoría de la selección natural, la evolución y la genética. Basándose de que es una mutación y no de un operador de cruce, y de que se usara la alteración biológica para mejorar la prueba inicial factible de la búsqueda tabú, ya que, de acuerdo con S.Hillier & J. Lieberman, (2010), esta metaheurística debe de comenzar con una solución de prueba inicial factible que, teóricamente mejorándola, puede igualar o mejorar la respuesta final del algoritmo.

Se ha determinado el conjunto de parámetros iniciales (longitud tabú, número

de iteraciones y número de diversificaciones) que hacen que el método tenga un mejor desempeño computacional para cada uno de los casos de prueba, calibrándolos por ensayo y error. (Johnson & McGeoch, 2018)

Se recomienda que la longitud tabú esté en el intervalo  $[\frac{n}{2} - 3, \frac{n}{2} + 3]$ , donde “n” representa la cantidad de nodos a visitar y que no sobrepasen más de 10 diversificación. (Johnson & McGeoch, 2018)

## 7.2. Marco conceptual

A continuación, se presentan algunas teorías relevantes con el fin de entender el algoritmo de búsqueda tabú y vecino más cercano, entre otros:

**Agente viajero:** Según Anaya Fuentes et al., (2016), es un problema de optimización combinatoria en la que una persona visita sólo en una ocasión cada una de las ciudades y se regresa al punto de partida; se deberá localizar la ruta que presente la distancia más corta y a ésta se la conoce como la ruta óptima. A medida que crece el número de ciudades a visitar por el agente viajero, no es factible resolverlo por programación entera debido a que el tiempo de solución computacional crece de forma exponencial de acuerdo con el número de ciudades visitadas; a este tipo de problemas se los conoce como no polinomiales (NP).

**Algoritmo:** De acuerdo con Mota Verdura, (2016), un algoritmo es un objeto A que satisface los postulados de secuenciación temporal, de estados abstractos y de exploración delimitada.

**Búsqueda Tabú:** Según Glover & Melian, (2003), es un procedimiento metaheurístico cuya característica distintiva es el uso de memoria adaptativa y de estrategias especiales de resolución de problemas. Su filosofía se basa en la explotación de diversas estrategias inteligentes para la resolución de problemas, basadas en procedimientos de aprendizaje. El marco de memoria adaptativa de TS explota la historia del proceso de resolución del problema haciendo referencia a cuatro dimensiones principales, consistentes en la propiedad de ser reciente, en frecuencia, en calidad, y en influencia.

**Cadena de frío:** Según Navarro, (2013), es el proceso de garantizar el mantenimiento de la temperatura de los productos perecederos a lo largo de fases sucesivas, de acuerdo con las características de cada producto.

De acuerdo con directrices del gobierno nacional, establecidas en la “guía para el monitoreo de temperaturas en el transporte de biológicos (Ministerio de Salud y protección social, 2018)”, los rangos de temperatura en los que pueden estar los biológicos sin adquirir consecuencias desastrosas oscilan entre +2 °C a +8 °C.

**Heurística:** Según Moreno Pérez, (2004), es aquella solución sobre la que se tiene cierta confianza de que es factible y óptima, o de que alcanza un alto grado de optimalidad y/o factibilidad.

**Inmunidad de rebaño:** De acuerdo con Paul & Dvm, (s/f) es la habilidad de un individuo para resistir o limitar la severidad de la infección o enfermedad y que depende de la naturaleza del agente, la dosis de desafío y el ambiente.

**Logística:** Según (*15 conceptos básicos de logística y cadena de suministro, s/f*), se refiere a los procesos de coordinación, gestión y transporte de los bienes comerciales desde el lugar de distribución hasta el cliente final. Por tanto, la logística se encarga del producto en cuanto a su almacenamiento, inventario, transporte, entrega y/o devolución.

**Logística de distribución:** De acuerdo con (*¿Qué es la Logística de distribución?, s/f*), es la planificación y el control del movimiento físico de materiales y productos terminados desde los lugares de fabricación hasta los sitios de consumo.

**Metaheurística:** De acuerdo con Moreno Pérez, (2004), define las metaheurísticas como las estrategias generales para construir algoritmos, que quedan por encima de las heurísticas, y van algo más allá.

**Mutación:** De acuerdo con Klug et al., (2012), define las mutaciones como una alteración de la secuencia de ADN, que pueden producir cambios dentro de las regiones no codificadas de un gen.

**Optimización combinatoria:** De acuerdo con Mejía Moncayo, (2012), son aquellas variables discretas donde las posibles soluciones son permutaciones o combinaciones de estas, que tienen un espacio de búsqueda finitos, que en algunos casos pueden tender a dimensiones infinitas.

**Ruta:** según (*Ruta de transporte: tipos, diseño y planificación de rutas logísticas, s/f*), es el recorrido que realiza el transportista durante la distribución y/o entrega de mercancía.

**Vecino más cercano:** Considerado como un heurístico según Cepeda et al., (s/f), es el algoritmo más simple y fácil de construir, pues nos permite encontrar la mínima distancias desde el depósito hacia cada uno de los clientes.

**Vacuna:** Según la Real academia española, (s/f), define vacuna como, el preparado de antígenos que, aplicado a un organismo, provoca en él una respuesta de defensa.

### 7.3. Marco legal

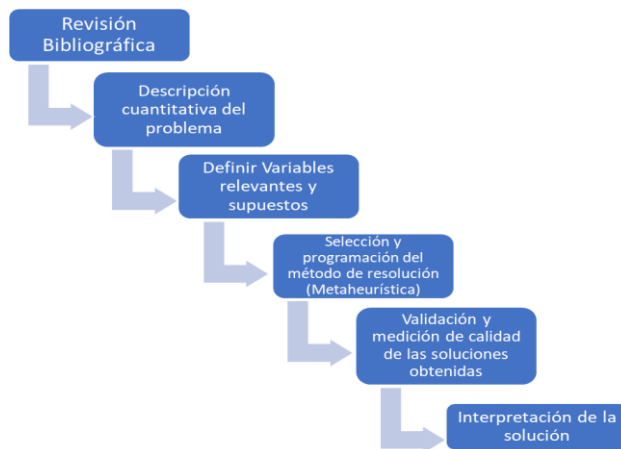
Las siguientes son algunas normativas legales aprobados por el gobierno nacional, que han ayudado en relevancia la realización de esta investigación:

<b>Norma o guía</b>	<b>Fecha</b>	<b>Aspecto legal</b>
Decreto 109 (Ministerio de Salud y protección social, 2021)	2021	El presente decreto tiene por objeto adoptar el plan nacional de vacunación contra el COVID-19 y establecer la población objeto, los criterios de priorización, las fases y la ruta para la aplicación de la vacuna, las responsabilidades de cada actor tanto del sistema general de seguridad social en salud como de los administradores de los regímenes especiales y de excepción, así como el procedimiento para el pago de los costos de su ejecución.
Guía para el monitoreo de temperaturas en el transporte de biológicos. Código: GIPG06 (Ministerio de Salud y protección social, 2018)	2018	El presente trabajo tiene como objetivo el monitoreo de las temperaturas durante el transporte de los biológicos a nivel nacional mediante el uso de registradores de datos al interior de los embalajes con el fin de garantizar la cadena de frío según las recomendaciones del fabricante.
Circular externa, Gobernación de Antioquia Radicado: K2021090000042 Fecha: 08/02/2021 (Salud et al., 2021)	2021	Posibles consecuencias del incumplimiento de la priorización dentro del plan nacional de vacunación contra el covid -19.

**Tabla 1**

## 8. METODOLOGÍA

La presente investigación es de tipo cuantitativo, por medio de dos modelos matemáticos con parámetros y variables numéricas se busca adaptar, a partir de algoritmos metaheurísticos, un algoritmo de resolución de rutas para el reparto de vacunas COVID-19 en la red hospitalaria de Antioquia. Las fases para desarrollar la metodología se muestran en la siguiente figura:



**Figura 1**

La metodología seleccionada para el desarrollo de esta investigación se basa de la literatura de S.Hillier & J. Lieberman, (2010).

### 8.1.Descripción cuantitativa del problema

- $N = 41$
- Numero de restricciones: 82
- Numero de Variables: 1681
- Numero de recorridos posibles:  $8,15915283247898 * 10^{47}$

### 8.2.Definición de las variables de decisión modelo 1 (matriz de distancias):

$X_{ij} = 1$  Si el vendedor tiene que ir del origen  $i$  al destino  $j$ , cero de lo contrario

$X_{ij} =$  Donde  $i$  y  $j$  toman valores desde 1,2,3 hasta 41

### 8.3.Definición de las variables de decisión modelo 2 (matriz de tiempos):

$X_{ij} = 1$  Si el vendedor tiene que ir del origen  $i$  al destino  $j$ , cero de lo contrario

$X_{ij} =$  Donde  $i$  y  $j$  toman valores desde 1,2,3 hasta 41

#### 8.4. Definición de la función objetivo modelo 2

$$\text{Min } Z = \sum_{i=1}^n \sum_{j=1}^n t_{ij} X_{ij} \quad t_{ij} = M \text{ si } i = j$$

M = Número muy grande

$t_{ij}$  = Matriz de tiempos del origen  $i$  al destino  $j$ , tomados en Google Maps en el mismo intervalo de tiempo para eliminar el factor de cambio en el tráfico.

#### 8.5. Definición de la función objetivo modelo 1

$$\text{Min } Z = \sum_{i=1}^n \sum_{j=1}^n d_{ij} X_{ij} \quad d_{ij} = M \text{ si } i = j$$

M = Número muy grande

$d_{ij}$  = Matriz de distancias del origen  $i$  al destino  $j$ , tomados en Google Maps en el mismo intervalo de tiempo para eliminar el factor de cambio en el tráfico.

#### 8.6. Restricciones del modelo:

Restricciones de Origen  $\sum_{i=1}^n X_{ij} = 1$  Donde  $i = 1, 2, \dots, n$

Restricciones de Destino  $\sum_{j=1}^n X_{ij} = 1$  Donde  $j = 1, 2, \dots, n$

$$X_{ij} = \text{Binaria } (0, 1)$$

#### 8.7. Definición de variables relevantes y supuestos

##### 8.7.1. Supuestos:

- El problema clásico del agente viajero (TSP) contempla una ruta cerrada, partiendo y terminando en el mismo punto y visitando exactamente una vez cada nodo.

- La matriz de distancias  $d_{ij}$  medida en kilómetros y de tiempos  $t_{ij}$  medida en minutos son asimétricas, es decir, no es lo mismo ir del punto A al punto B que del punto B al punto A.
- Para la realización del algoritmo no se tendrá en cuenta el número de biológicos por camión (Restricción de capacidad del vehículo), partiendo del supuesto que un camión puede transportar la cantidad suficiente para abastecer a los hospitales de la ruta. Tampoco se contempla la hora de salida de los vehículos y el tipo de camiones usados. Tampoco se contemplan ventanas de tiempo para los hospitales que reciben las vacunas. No se tienen en cuenta tiempos de cargue y descargue de vacunas en ninguno de los puntos de la ruta, el objetivo de la investigación es determinar la secuencia que minimice la distancia total recorrida o el tiempo total de la ruta, más no la ejecución operativa del proceso de cargue y descargue de biológicos.
- Debido a cuestiones de seguridad por parte del gobierno nacional, la ubicación de la bodega departamental en Antioquia no pudo ser localizado con exactitud, por lo tanto, basándose de información oficial de la gobernación de dicho departamento, el recorrido deberá empezar y terminar en la calle 38 #55-310, Bello (Hospital mental de Antioquia).
- La fuente de datos de la matriz de distancias y tiempos es Google Maps. Si la aplicación muestra más de una ruta posible se selecciona la de menor distancia para el Modelo 1 y la de menor tiempo para el modelo 2. Los datos de distancias y tiempos entre los diferentes orígenes y destinos fueron tomadas en la misma franja horaria, esto para eliminar como factor de cambio las posibles alteraciones de tráfico o cierre de vías. La presente investigación supone que seleccionar una ruta respecto de otra no depende del tráfico puesto que todas las rutas parten de las mismas condiciones de congestión vial. Si existen cambios considerables en la disponibilidad de vías, como las vividas actualmente a causa de la contingencia social se debe volver a tomar tiempos y distancias para contemplar dichas alteraciones.
- El cálculo de costos del Modelo 2, donde la ruta solución es la ruta con menor tiempo, se determina por medio del cálculo de la distancia total de dicha ruta solución y posteriormente se calcula el costo de la misma.

## 8.8. Variables relevantes

De acuerdo con el DANE, el Índice de Costos del Transporte de Carga por Carretera (ICTC) permite medir la variación porcentual promedio de precios de un conjunto representativo de bienes y servicios necesarios para garantizar la movilización de un vehículo prestador del servicio del transporte de carga por carretera en el país a lo largo del tiempo. Es un indicador

especializado que permite la toma de decisiones por parte de entidades del gobierno y empresas privadas del sector.

Dentro de la metodología que utiliza el DANE para el cálculo del ICTC-15 se contemplan costos relacionados con: salarios y costos sociales, servicios de reparación, combustible, desgaste de llantas, costos de administración, costos de aseguramiento e impuestos, peajes y costos de capital.

Para la presente investigación se utiliza el ICTC-15 correspondiente a Enero de 2021 como referencia del costo promedio por kilómetro recorrido, dando claridad que el costo de transporte de carga que requiere control de temperatura puede ser mayor debido a los costos adicionales no contemplados en la metodología utilizada por el DANE para determinar este indicador.

Con el fin de tener un acercamiento más aproximado en términos económicos, el precio por kilómetro recorrido se basa de un simulador del ministerio de transporte donde establece el siguiente precio en el Valle de Aburrá: \$ 11.821,13.

## 8.9. Selección y programación del método de resolución (Metaheurística)

### 8.9.1. Selección de la mejor metaheurística

Para el desarrollo de la presente investigación se tomó como punto de partida, los resultados de un proyecto de investigación formativa PIF, de la materia metaheurística, que consistió en emplear tres algoritmos (Algoritmo genético, Recocido simulado y Búsqueda tabú) para dar resolución a un problema clásico del agente viajero de  $N = 16$ , y examinar la calidad de la mejor solución frente a las demás estrategias:

Tiempos

Metaheurística	Mejor Solución	Función Objetivo	Ranking	Indicador de calidad
Búsqueda Tabú	0-6-13-9-4-14-7-10-5-15-1-12-2-3-8-11-0	129	1	Mejor solucion
Recocido Simulado	0-14-13-9-4-6-5-10-7-12-1-15-11-3-8-2-0	131	2	98,4496124
Algoritmos Genéticos	0-13-6-9-4-10-12-1-15-11-3-8-5-2-14-7-0	168	3	69,76744186

Fuente: Elaboración propia

**Tabla 2**

## Distancias

Metaheurística	Mejor Solución	Función Objetivo	Ranking	Indicador de calidad
Búsqueda Tabú	0-6-13-9-4-14-5-10-7-2-3-8-12-1-15-11-0	29,72	1	Mejor solución
Recocido Simulado	0-6-13-9-4-14-5-10-7-12-1-15-8-3-11-2-0	31,62	2	93,60699865
Algoritmos Genéticos	0-10-13-6-9-2-8-3-12-1-14-15-5-4-7-11-0	40,62	3	63,3243607

Fuente: Elaboración propia

**Tabla 3**

Basándose de los resultados anteriores, se determinó que el mejor algoritmo para trabajar la problemática de la presente investigación es **búsqueda tabú**, ya que, esta metaheurística obtuvo las mejores soluciones en términos de tiempo y distancias.

### 8.9.2. Programación del método de solución

Para lograr programar el método de solución, se procedió inicialmente a la toma de datos, mediante las herramientas de Google maps y Excel para Microsoft 365 MSO de 64 bits, dejando como resultado, la siguiente muestra visible de las tablas originales:

Modelo 1 (Matriz de distancias dij)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	M	16,1	10,8	3	5,3	16	17,4	16,2	13,9	18,1	13,1	12,3	14,4	14,4
1	14,2	M	7,9	12	10,2	5,3	11,9	4,9	1,9	2,8	14	7,3	7,2	8,4
2	11,9	8,7	M	11,1	8,5	10,4	12,1	9,8	7,5	11,7	3,7	2,3	4,2	4,9
3	3,4	11,6	9	M	2,3	10,1	10,7	14,2	11,8	16,2	10,9	10	12,1	12,2
4	5,5	9,4	5,6	3	M	10	10,3	11,5	8,9	13,8	8	7,2	9,2	9,3
5	15,8	5,1	9,5	10,1	11,8	M	6,5	15,1	5,7	6,9	7,5	8,8	8,7	9,9
6	17,2	11	11,7	11,6	12,5	6,4	M	15,1	12,4	12,8	12	12	13,2	14,4
7	16,6	5	9,7	14,7	12,6	15,3	15,3	M	4,4	2,5	7,6	8,9	8,8	10
8	14,1	2,5	7,4	10,7	8,6	5,8	11,6	4,2	M	4,8	5,5	7,4	6,7	7,9
9	16,1	3,1	11,9	14,2	12,1	7,3	13,8	2,8	4,2	M	7,9	9,2	9,1	10,3
10	13,4	6,7	3,5	11,5	9,4	7,6	12,3	7,9	5,5	9,8	M	2,2	1,2	3,1
11	11,9	8,7	2,3	9,8	7,7	9,8	12,1	9,8	7,5	11,7	2,4	M	3	2,1
12	13,8	7,9	4,4	12,7	10,6	8,7	13,4	9	6,6	10,9	1,1	3,1	M	2,8
13	13,9	9,1	4,5	13,1	9,7	10	14,1	10,3	8	12,1	3,2	2,1	3,3	M
14	13,6	10,4	4,5	11,5	9,4	11,5	13,8	11,5	9,1	13,4	3,7	1,9	3,7	1,1
15	13,3	5,3	4,1	11,4	9,3	7,5	12,2	7,8	4	9,7	1,2	2,8	2,7	3,9
16	5,5	9	5,8	8,7	6,6	14,4	15,1	15,9	13,3	17,7	9,3	7,9	9,9	10,5
17	8,5	9,8	2,9	6,3	4,2	12	14,3	12,3	9,8	15,3	6,5	5,2	7,1	7,7
18	10,8	10,9	1,6	9	6,9	11	12,4	10,9	8,6	12,8	5,2	3,3	5,7	5,9
19	7,7	7,8	3	5,9	3,8	8,9	9,3	10,4	7,9	12,3	6,1	4,3	7,3	6,9
20	6,8	9,2	3,5	4,6	2,5	10,3	12,6	11,8	9,3	13,7	9,3	7,8	10,5	8,4

**Tabla 4**

## Modelo 2 (Matriz de tiempos $t_{ij}$ )

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	M	28	23	11	11	28	30	25	23	29	24	23	27	29
1	29	M	18	28	24	15	24	11	5	5	6	18	16	22
2	26	17	M	23	16	23	26	17	16	23	14	7	16	17
3	12	24	22	M	7	25	24	23	21	27	23	22	25	29
4	18	17	17	10	M	20	19	16	15	20	16	15	19	22
5	33	15	30	28	21	M	17	23	14	16	17	21	20	25
6	34	23	27	26	21	18	M	22	22	23	22	21	24	28
7	31	11	21	27	18	25	24	M	11	7	16	20	18	24
8	30	2,5	19	25	17	15	22	9	M	12	15	19	19	24
9	33	9	26	29	20	18	26	7	10	M	20	24	23	28
10	28	15	14	25	16	18	21	15	15	20	M	8	5	11
11	25	18	7	22	14	20	22	18	17	23	9	M	10	9
12	13	19	17	29	20	22	25	20	19	24	4	11	M	11
13	33	26	17	30	22	27	28	26	25	30	14	10	12	M
14	36	25	17	32	22	27	29	26	26	32	18	9	16	6
15	31	14	13	23	15	16	21	14	12	19	6	11	10	15
16	17	22	26	33	25	36	37	39	38	43	35	29	34	34
17	26	22	9	20	12	24	26	26	24	30	23	16	22	24
18	33	22	5	27	19	25	26	22	21	27	18	10	18	18
19	20	15	11	15	8	17	17	16	15	20	17	15	18	24
20	19	16	12	14	6	18	20	18	16	22	19	16	20	26

**Tabla 5**

### 8.9.3. Método de solución, algoritmo clásico de Búsqueda Tabú

Según la literatura de Frederick S. Hillier, Gerald J. Lieberman, 2010, el paso inicial del algoritmo clásico es comenzar con una ruta inicial factible, que, para este caso se usa una heurística llamada el “vecino más cercano, cuyos pasos son los siguientes:

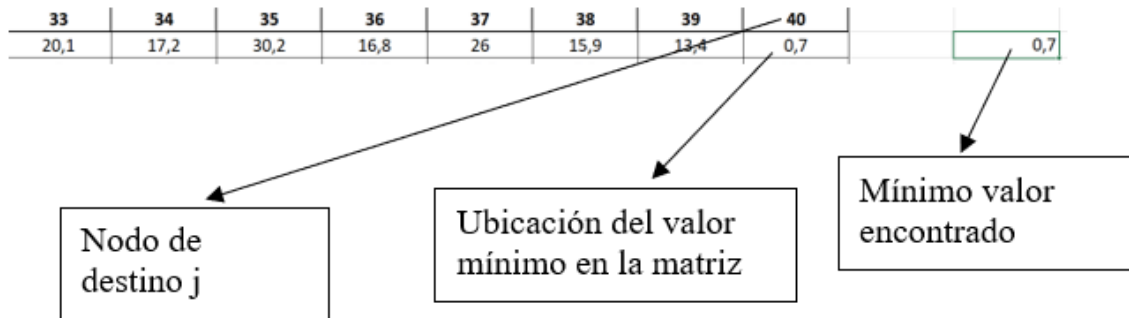
**Paso 1:** Se inicia ubicándose en la matriz de su preferencia desde el origen  $i$  en el nodo cero.

	0	1	2	3	4	5	
0	0	M	16,1	10,8	3	5,3	16

**Figura 2**

**Paso2:** Una vez ubicados, nos desplazamos a lo largo de la fila del origen  $i$ , hasta alcanzar el mínimo valor en algún nodo del destino  $j$ .

33	34	35	36	37	38	39	40	
20,1	17,2	30,2	16,8	26	15,9	13,4	0,7	=MIN(C4:AQ4)



**Figura 3**

En este momento encontramos que el valor más cercano partiendo desde cero está en el nodo de destino 40, es decir que la ruta inicial factible se encuentra así: 0-40.

**Paso 3:** Marcamos toda la columna del nodo de destino 40, con el fin de evitar seleccionarla en las siguientes iteraciones.

	31	32	33	34	35	36	37	38	39	40	
	14,9	15	20,1	17,2	30,2	16,8	26	15,9	13,4	0,7	0,7
	4,4	6,2	8	4,5	24,7	15,5	15,5	5,8	8,7	13,1	
	7,1	9,8	14,8	10,3	24,9	11,5	20,8	6,7	13,6	10,8	
	10,1	9	13,5	11,1	23,5	8,1	23,8	13,7	11,2	2,6	
	6,8	8,9	11,8	6	21,9	8,5	20,9	10,8	8,3	4,3	
	4,7	2,1	4,4	1,8	20,1	6,7	20,2	10	10,2	14,6	
	9,3	7,2	2,8	11,3	13,1	3	25,2	15	13,1	16	
	8,8	12,3	18,1	8,3	28,1	14,7	10,9	3,3	10,8	15,5	
	4,3	6,6	8,4	6	24,3	11	14,2	3,9	8,4	12,9	
	6,3	8,1	10,4	6,4	26,6	13,2	13,3	5,4	10,2	15	
	5,6	8,4	15	8,1	25	11,7	18,2	5,1	3,2	12,3	
	7,1	9,7	14,8	11,1	24,9	11,5	20,8	6,5	1,3	10,8	
	6,7	9,5	16,1	9,2	26,2	12,8	19,2	6	3,6	12,6	

**Figura 4**

**Paso 4:** Repetimos los pasos del 1 – 3 hasta que ya no queden más nodos y se tenga la ruta inicial factible.

Una vez tengamos las rutas iniciales factibles de tiempo y distancias, procedemos a la programación de la macro en Excel, usando el lenguaje visual Basic (la cual está configurada para procesar la matriz de 41\*41) que nos ayudara a procesar los más de 1600 datos:

### 8.10. Explicación de la metodología empleada en la programación:

La programación de la macro en visual Basic, parte de la siguiente metodología establecida en la literatura por Frederick S. Hillier, Gerald J. Lieberman, (2010) que instaura los pasos para el sub-viaje inverso y la búsqueda tabú y cuyo código se encuentra en el anexo 1.

#### Reglas del sub-viaje inverso

**Paso inicial:** inicie con cualquier viaje factible como la solución de prueba inicial.

**Iteración:** Para la solución de prueba actual, considere todas las formas posibles de realizar un sub-viaje inverso (sólo excluya el inverso del viaje completo caso simétrico). Seleccione aquel que proporcione la mayor disminución en la distancia viajada para que sea la nueva solución de prueba. (Los empates se rompen de manera arbitraria).

**Regla de detención:** El proceso se detiene cuando ninguna inversión de un sub-viaje mejora la solución de prueba actual, la cual se acepta como la solución final.

### 8.11. Pasos con ejemplo tomados de (S.Hillier & J. Lieberman, 2010):

**Paso 1:** Solución de prueba inicial es 1 2 3 4 5 6 7 1 la distancia es 69 km

**Paso 2:** Considerar todas las posibles formas de realizar sub-viaje inverso

Sub-viaje inverso de 2

Inverso	Ruta
2-3	1 3 2 4 5 6 7 1
3-4	1 2 4 3 5 6 7 1
4-5	1 2 3 5 4 6 7 1
5-6	1 2 3 4 6 5 7 1

Fuente :(S.Hillier & J. Lieberman, 2010)

**Tabla 6**

Lo que hace el sub-viaje inverso es voltear los nodos sin tener en cuenta la partida (1), con el fin de que las diferentes combinaciones reduzcan aún más la función objetivo.

**Paso 3:** Considerar todas las posibles formas de realizar subviaje inverso Solución de prueba actual 1 2 4 3 5 6 7 1 con distancia de 65 km

### Sub-viaje inverso de 3

Inverso	Ruta
2-4-3	1 3 4 2 5 6 7 1
4-3-5	1 2 5 4 3 6 7 1
3-5-6	1 2 4 6 5 3 7 1
5-6-7	1 2 4 3 7 6 5 1

Fuente: (S.Hillier & J. Lieberman, 2010)

**Tabla 7**

Como se observa en la tabla anterior, en el sub-viaje inverso de 3, cambia de posición los números en rojo y los de color negro permanecen quietos, cada 3 posiciones.

**Paso 4:** Considerar todas las posibles formas de realizar sub-viaje inverso solución de prueba actual 1 2 4 6 5 3 7 1 con distancia de 64 km

### Sub-viaje inverso de 3

Inverso	Ruta
2-4-6	1 6 4 2 5 3 7 1
4-6-5	1 2 5 4 3 6 7 1
6-5-3	1 2 4 6 5 3 7 1
5-3-7	1 2 4 3 7 6 5 1

Fuente: (S.Hillier & J. Lieberman, 2010)

**Tabla 8**

Solución 1 2 4 6 5 3 7 1 con 64 Km.

#### **Pasos de la búsqueda tabú:**

##### 1. Algoritmo de búsqueda local:

En cada iteración seleccione el mejor vecino inmediato de la solución de prueba actual que no esté descartado por la lista tabú.

##### 2. Estructura de vecindad:

Un vecino inmediato de la solución de prueba actual es aquel al que se llega por medio de un sub-viaje inverso. Tal inversión requiere la adición de dos arcos y la eliminación de otros dos arcos de la solución de prueba actual. (Se descarta un sub-viaje inverso que sólo invierta la dirección del viaje que proporciona la solución de prueba actual para el caso simétrico.)

### 3. Forma de los movimientos tabú:

Enumere los arcos de forma que un sub-viaje inverso sea tabú si los dos arcos que se eliminan por esta inversión se encuentran en la lista. (Ello evitará que se regrese con rapidez a la solución de prueba anterior.)

### 4. Adición de un movimiento tabú:

En cada iteración, después de elegir los dos arcos que deben agregarse a la solución de prueba actual, también incorpore estos dos arcos a la lista tabú.

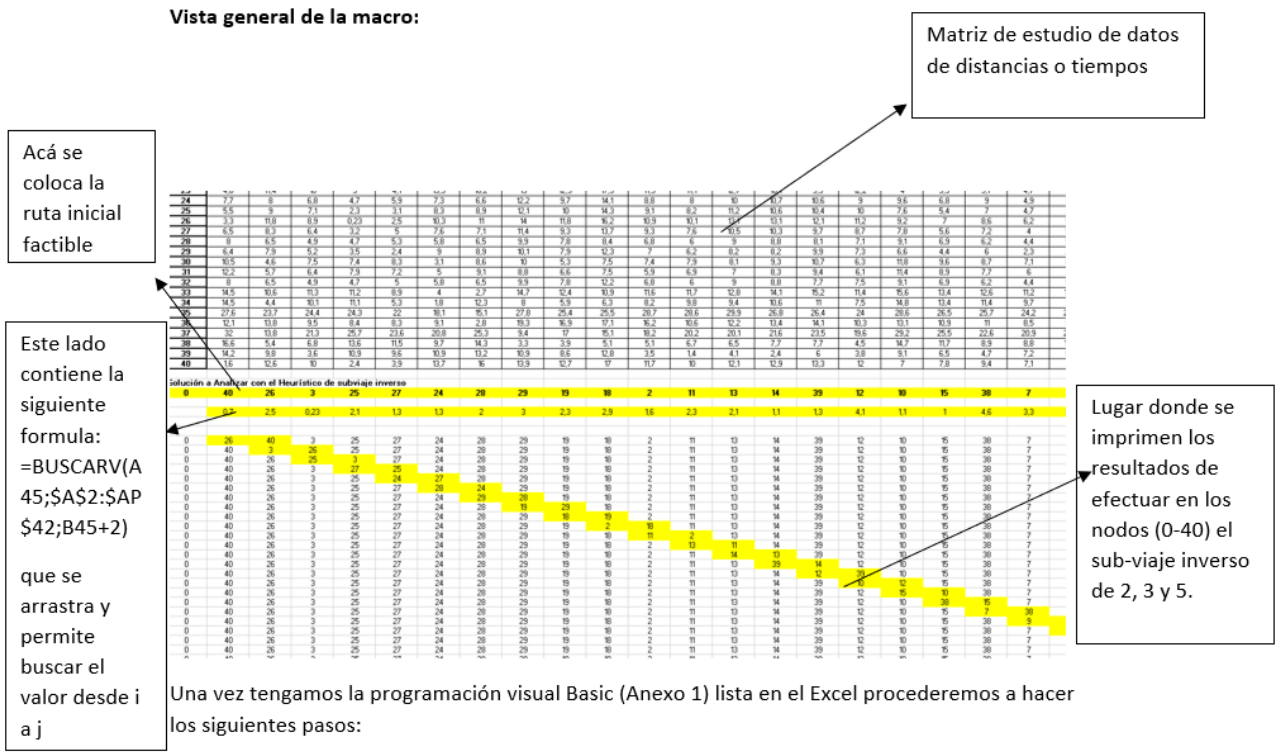
### 5. Tamaño máximo de la lista tabú:

Cuatro (dos de cada una de las dos iteraciones más recientes). Siempre que se agregue un par de arcos a una lista llena, elimine los dos arcos que han permanecido por más tiempo en la lista.

### 5. Regla de detención:

Detenga el proceso después de tres iteraciones consecutivas sin mejora del valor de la función objetivo. (También debe detenerse en cualquier iteración donde la solución de prueba actual no tenga vecinos inmediatos porque están penalizados por la lista tabú.)

## 8.12. Pasos de la metodología búsqueda tabú con sub-viaje inverso en macro Excel:



**Figura 5**

**Paso1:** Colocar la ruta inicial factible dada por el vecino más cercano

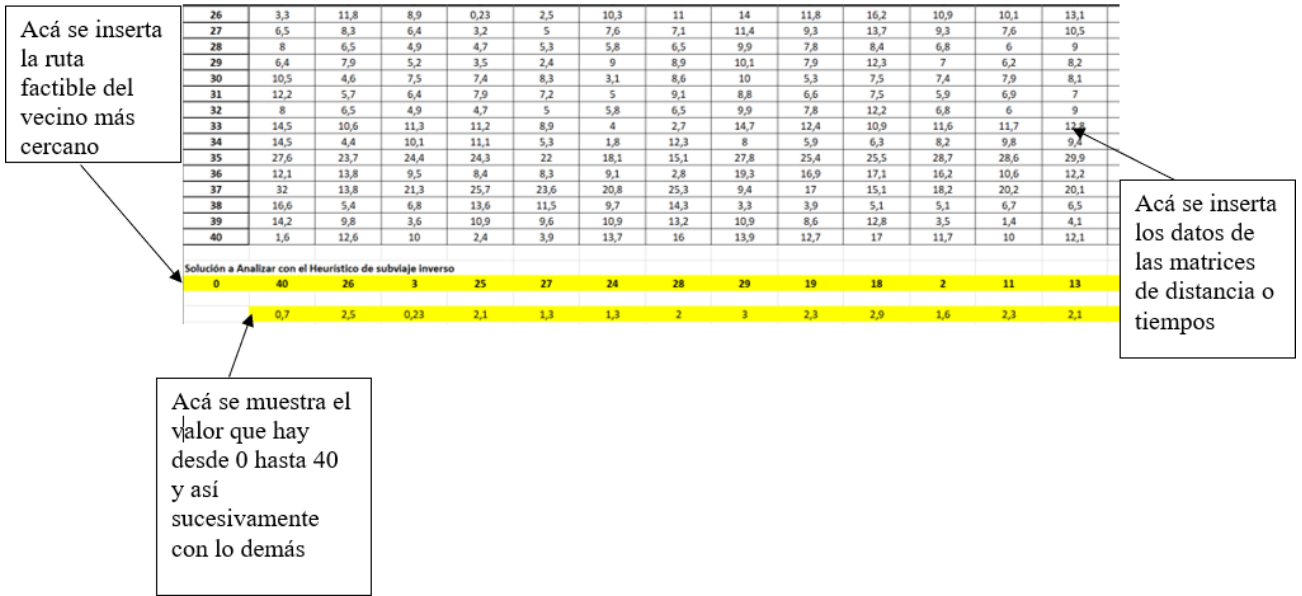


Figura 6

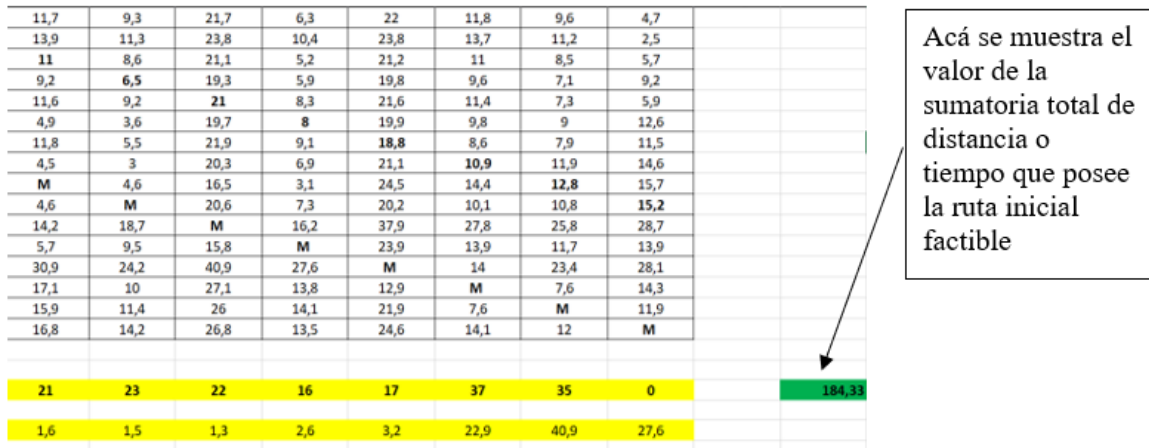
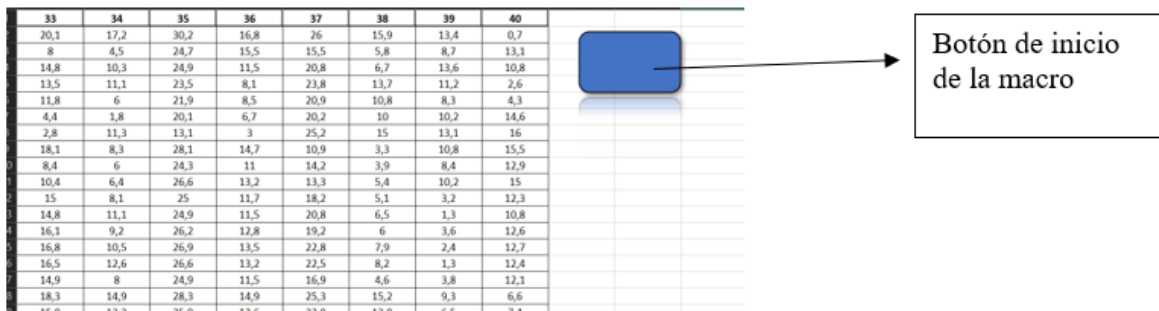


Figura 7



**Paso 2:** una vez con que se tengan los **Figura 8** lugares correspondientes, se procede a iniciar la macro con el botón respectivo previamente configurado.

Cuando se le da inicio la macro, aparecerá la siguiente ventana:

33	34	35	36	37	38	39	40
20,1	17,2	30,2	16,8	26	15,9	13,4	0,7
8	4,5	24,7	15,5	15,5	5,8	8,7	13,1
14,8	10,3	24,9	11,5	20,8	6,7	13,6	10,8
11,5	11,1	23,5	8,1	23,8	13,7	11,2	2,6
11,8	6	21,9	8,5	20,9	10,8	8,3	4,3
4,4	1,8	20,1	6,7	20,2	10	10,2	14,6
2,8	11,3	13,1	3	25,2	15	13,1	
18,1	8,3	28,1	14,7	10,9	3,3	10,8	
8,4	6	24,3	11	14,2	3,9	8,4	
10,4	6,4	26,6	13,2	13,3	5,4	10,2	
15	8,1	25	11,7	18,2	5,1	3,2	
14,8	11,1	24,9	11,5	20,8	6,5	1,3	
16,1	9,2	26,2	12,8	19,2	6	3,6	
16,8	10,5	26,9	13,5	22,8	7,9	2,4	12,7

Acá en esta ventana emergente se deberá colocar el primer dato de la ruta factible después del nodo cero (el nodo cero no se coloca) y dar enter, y así sucesivamente hasta completar los 40 datos.

Ojo:

En total hay 42 nodos contando el cero de inicio y el de llegada, pero NO se ingresará el cero de inicio y llegada en ninguna de las ventanas

Figura 9

**Paso 3:** Una vez halla procesado la macro buscaremos en esta sección la ruta más corta que nos muestra el algoritmo:

Para programar esta sección en el Excel, es de esta manera:

Administrador de reglas de formato condicional

Mostrar reglas de formato para: Selección actual

Nueva regla... Editar regla... Eliminar regla Regla duplicada

Regla (aplicada en el orden mostrado)	Formato	Se aplica a	Detener si es verdad
3 inferior	AaBbCcYyZz	= \$AR\$50:\$AR\$199	<input type="checkbox"/>

Aceptar Cerrar Aplicar

Proceso paso3

Figura 10

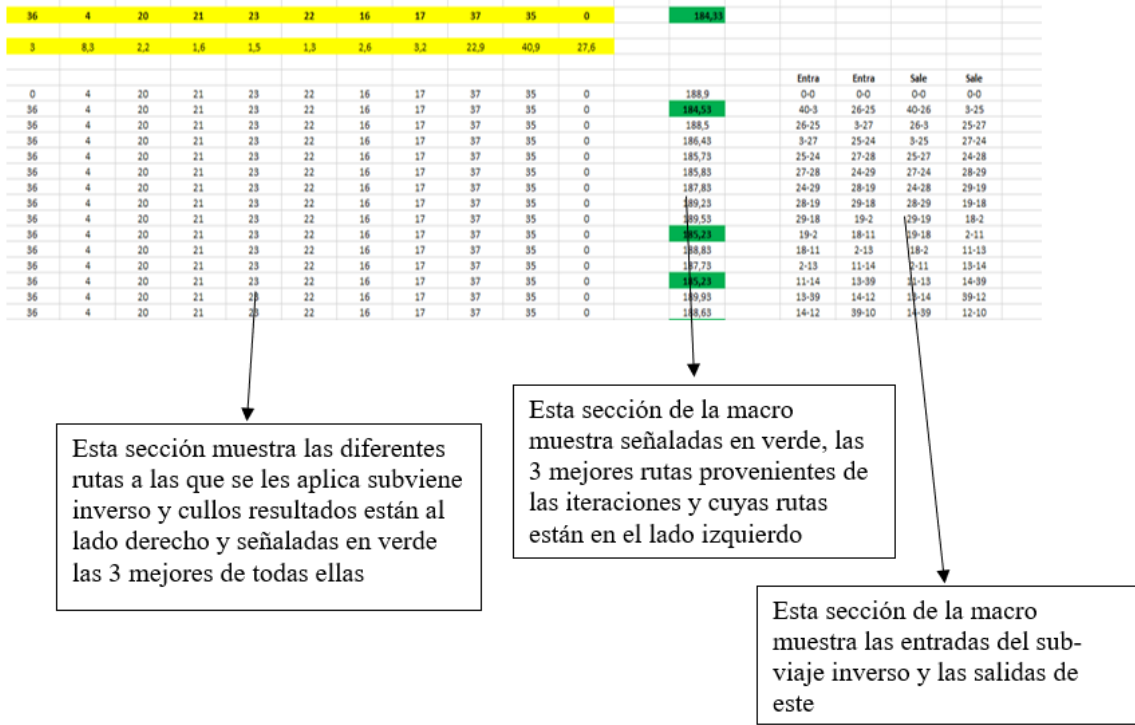


Figura 11

**Paso 4:** Una vez se tiene la ruta más corta, procederemos a iniciar la lista tabú la cual tendrá un máximo de 4.

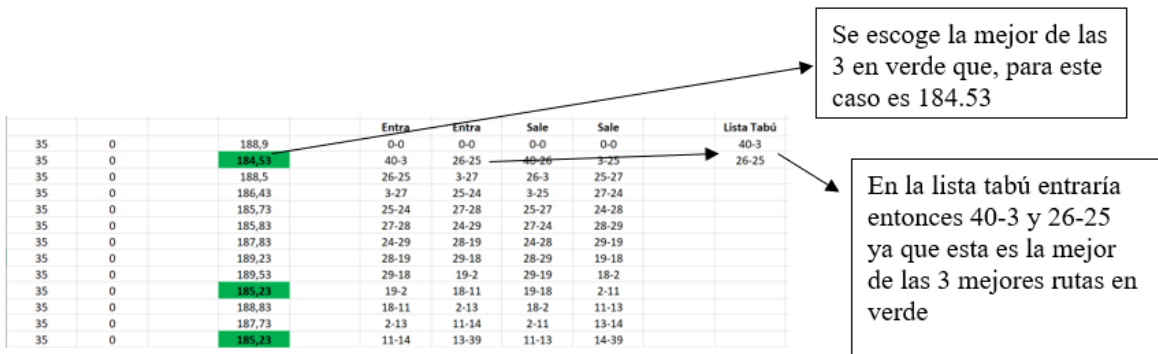


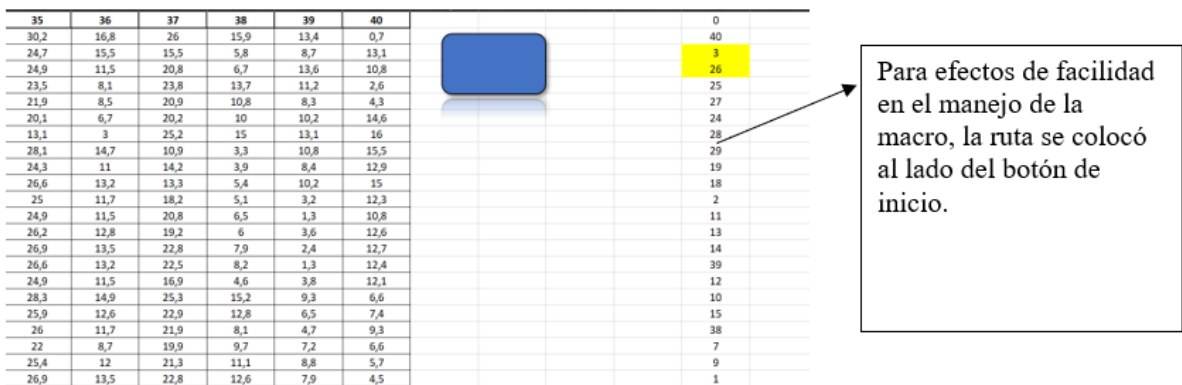
Figura 12

**Paso 5:** Una vez se tiene la lista tabú llena, se procederá a ubicar la ruta factible ganadora que para este ejemplo logro un resultado de 184.53



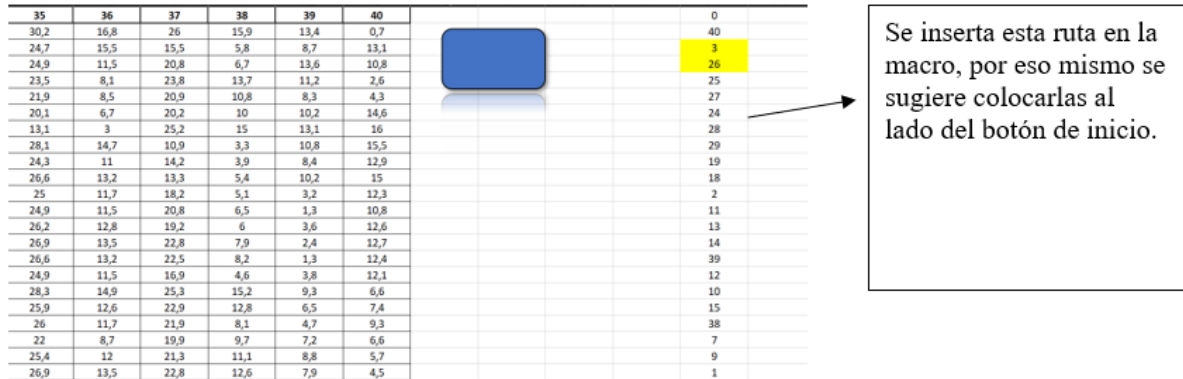
**Figura 13**

**Paso 6:** Una vez ubicada la ruta la copia y pegamos en una sección que permita su visualización con facilidad (Esto debido a que cuando iniciamos la macro, esta no permite el libre movimiento de la hoja de Excel).



**Figura 14**

**Paso 6:** Una vez ubicada la ruta factible, se vuelve a iniciar la macro, en donde aparecerá el letrero del paso 2. Una vez que aparezca el letrero se procede a los nodos de la ruta factible del paso 5.



**Figura 15**

**Paso 7:** Una vez allá terminado la ejecución del algoritmo, se procederá a buscar la mejor ruta entre las mejores señaladas en verde (cabe resaltar que puede que el código usado para marcar las mejores rutas en verde puede marcar incluso más de las estipuladas en la programación).

36	4	20	21	23	22	16	17	37	35	0	189,3
36	4	20	21	23	22	16	17	37	35	0	184,33
36	4	20	21	23	22	16	17	37	35	0	188,4
36	4	20	21	23	22	16	17	37	35	0	186,63
36	4	20	21	23	22	16	17	37	35	0	185,93
36	4	20	21	23	22	16	17	37	35	0	186,03
36	4	20	21	23	22	16	17	37	35	0	188,03
36	4	20	21	23	22	16	17	37	35	0	189,43
36	4	20	21	23	22	16	17	37	35	0	189,73
36	4	20	21	23	22	16	17	37	35	0	185,43
36	4	20	21	23	22	16	17	37	35	0	189,03
36	4	20	21	23	22	16	17	37	35	0	187,93
36	4	20	21	23	22	16	17	37	35	0	185,43
36	4	20	21	23	22	16	17	37	35	0	190,13
36	4	20	21	23	22	16	17	37	35	0	188,83
36	4	20	21	23	22	16	17	37	35	0	184,93

Para este ejemplo gano esta ruta ya que es la menor de todas

**Figura 16**

**Paso 8:** Se repite lo estipulado en el paso 4. Para este ejemplo ya tendríamos los 4 movimientos permitidos de la lista.

		Entra	Entra	Sale	Sale	Lista Tabú
189,3		0-3	40-26	0-40	3-26	40-3
184,33		40-26	3-25	40-3	26-25	26-25
188,4		3-25	26-27	3-26	25-27	40-26
186,63		26-27	25-24	26-25	27-24	3-25
185,93		25-24	27-28	25-27	24-28	

**Figura 17**

**Paso 9:** Se repiten los pasos 5, 6 y 7. Una vez ejecutados se vuelve a tener una ruta factible, pero con la lista tabú llena. Para este paso se procede a reemplazar la lista tabú por el entrante.

						Entra	Entra	Sale	Sale	Lista Tabú
16	17	37	35	0	188,9	0-26	40-3	0-40	26-3	40-3
16	17	37	35	0	184,53	40-3	26-25	40-26	3-25	26-25
16	17	37	35	0	188,5	26-25	3-27	26-3	25-27	40-26
16	17	37	35	0	186,43	3-27	25-24	3-25	27-24	3-25
16	17	37	35	0	185,73	25-24	27-28	25-27	24-28	
16	17	37	35	0	185,83	27-28	24-29	27-24	28-29	
16	17	37	35	0	187,83	24-29	28-19	24-28	29-19	
16	17	37	35	0	189,23	28-19	29-18	28-29	19-18	
16	17	37	35	0	189,53	29-18	19-2	29-19	18-2	
16	17	37	35	0	185,23	19-2	18-11	19-18	2-11	
16	17	37	35	0	188,83	18-11	2-13	18-2	11-13	
16	17	37	35	0	187,73	2-13	11-14	2-11	13-14	
16	17	37	35	0	185,23	11-14	13-39	11-13	14-39	
16	17	37	35	0	189,93	13-39	14-12	13-14	39-12	
16	17	37	35	0	188,63	14-12	39-10	14-39	12-10	
16	17	37	35	0	184,73	39-10	12-15	39-12	10-15	
16	17	37	35	0	185,83	12-15	10-38	12-10	15-18	

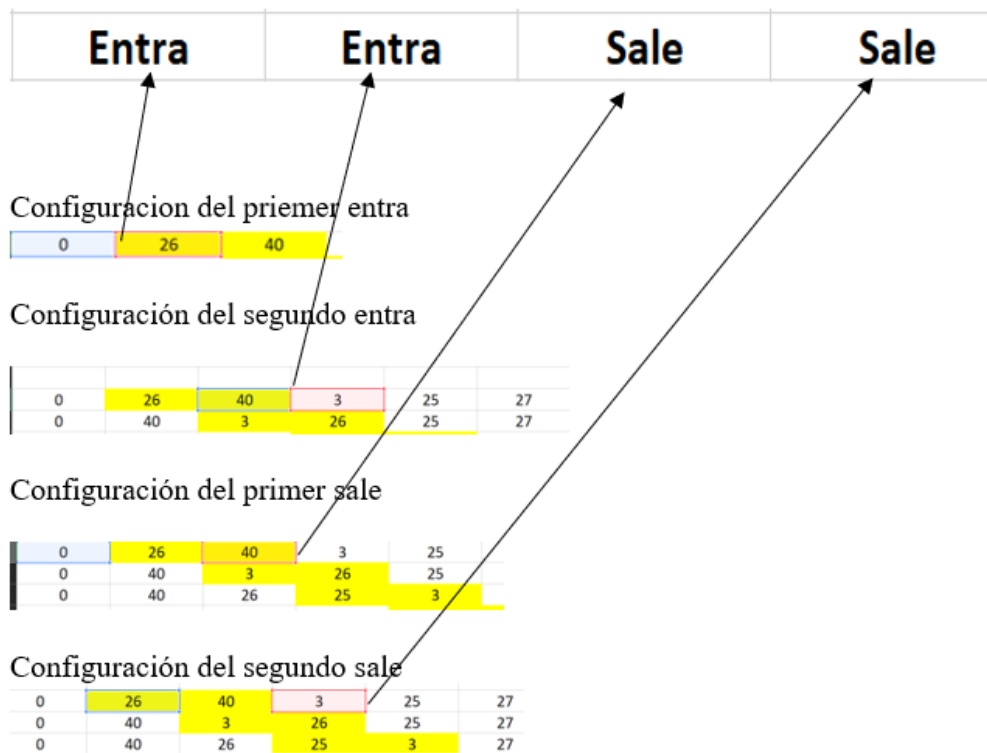
Cabe resaltar que para programar la sección de “entra” y “sale” es así:

Como se observa en las pestañas “sale” son iguales en la lista tabú, por lo tanto, esta ruta NO se coge, al no cogerse pasamos a la segunda mejor factible

**Figura 18**

Mediante la siguiente configuración en Excel configuramos las entra y sale de la macro, pero cada una tiene una ubicación especial

=CONCATENAR(A50;"-";B50)



**Figura 19**

Continuando con el paso 9, al no poder escoger la ruta factible por estar en la lista tabú seleccionamos la segunda mejor y se procede a hacer los pasos 6 y 7

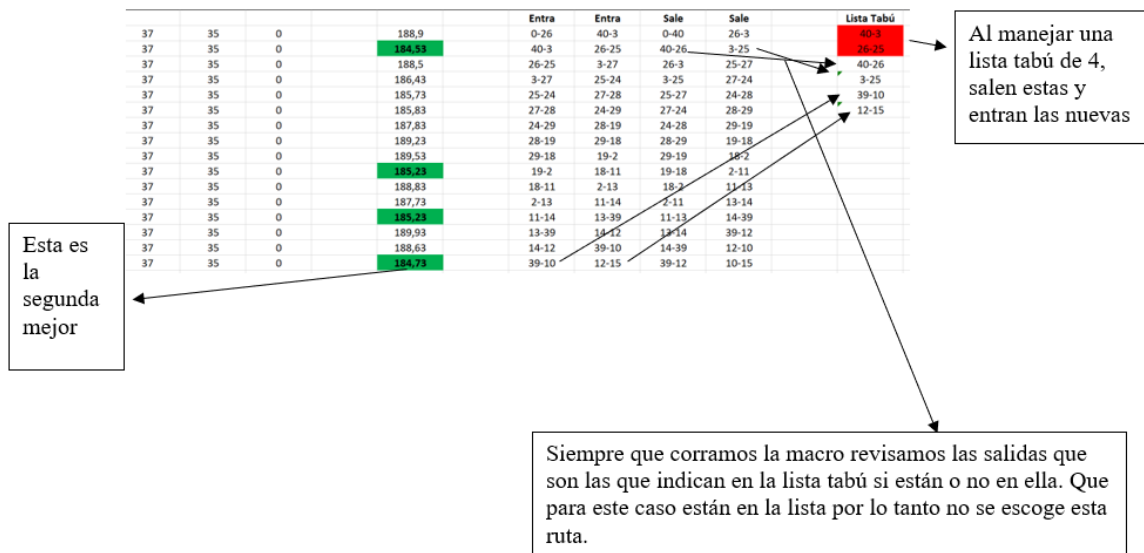


Figura 20

**Paso 10:** Repetimos los pasos del 2 – 9, hasta que encontremos la mejor ruta. Cabe resaltar que el algoritmo de búsqueda tabú permite desmejorar la mejor solución para después mejorar. Una vez que tengamos la mejor solución repetimos los pasos del 2 – 9 hasta llegar a la mejor solución 3 veces y es donde paramos.

### 8.13. Método de mutación adaptada

**Paso 1:** Aplicamos el heurístico el vecino más cercano (en ambas matrices de distancia y tiempo) con el fin de obtener las rutas iniciales factibles.

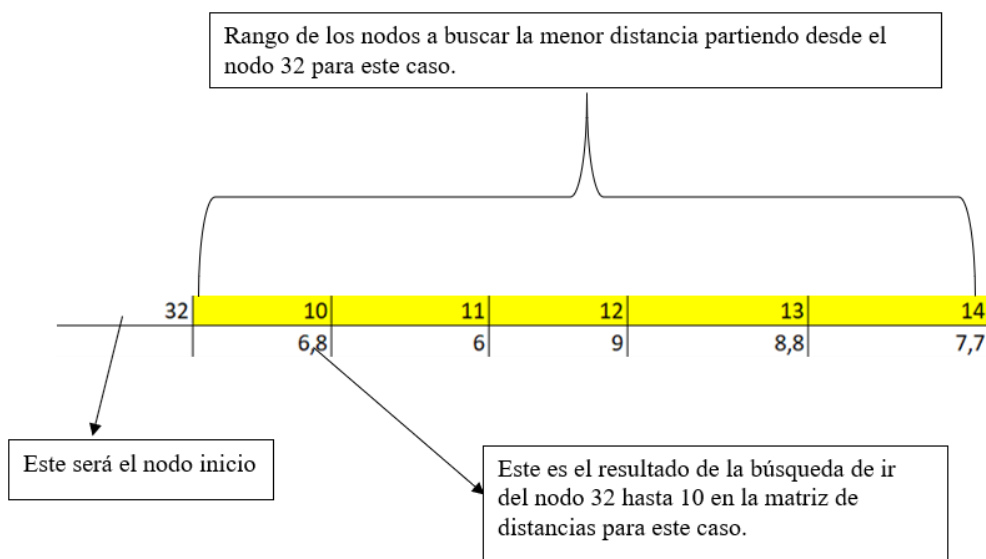
**Paso 2:** Una vez se tienen las rutas iniciales factibles se procede a ordenar estas en forma de doble hélice vertical desdoblada de la siguiente manera:

0	-	0	-	0	-	0	-	0
40	-	40	-	40	-	40	-	40
3	-	3	-	4	-	4	-	4
26	-	26	-	29	-	29	-	29
25	-	25	-	20	-	20	-	20
27	-	27	-	21	-	21	-	21
24	-	24	-	22	-	22	-	22
28	-	28	-	23	-	23	-	23
29	-	29	-	26	-	26	-	26
19	-	19	-	3	-	3	-	3
18	-	18	-	25	-	25	-	25
2	-	2	-	27	-	27	-	27
11	-	11	-	24	-	24	-	24
39	-	39	-	28	-	28	-	28
13	-	13	-	30	-	30	-	30
14	-	14	-	32	-	32	-	32
10	-	10	-	19	-	19	-	19
15	-	15	-	2	-	2	-	2
12	-	12	-	18	-	18	-	18
38	-	38	-	17	-	17	-	17
7	-	7	-	16	-	16	-	16
9	-	9	-	1	-	1	-	1
1	-	1	-	8	-	8	-	8

**Figura 21**

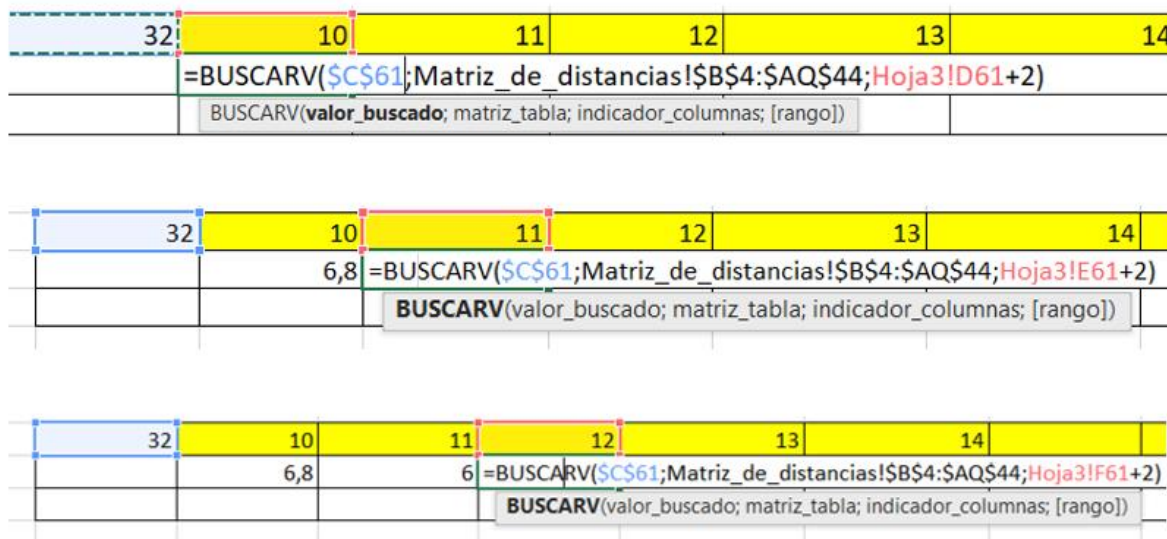
Para este caso el orden es irrelevante en el sentido de que las rutas iniciales factibles de distancia y tiempo pueden estar ubicados en la derecha e izquierda o de la forma contraria.

**Paso3:** Se programa una pequeña tabla que permita buscar en las matrices la distancia o el tiempo respectivo de cada uno, desde un origen especifico el cual va a estar anclado y no se moverá de la siguiente manera:



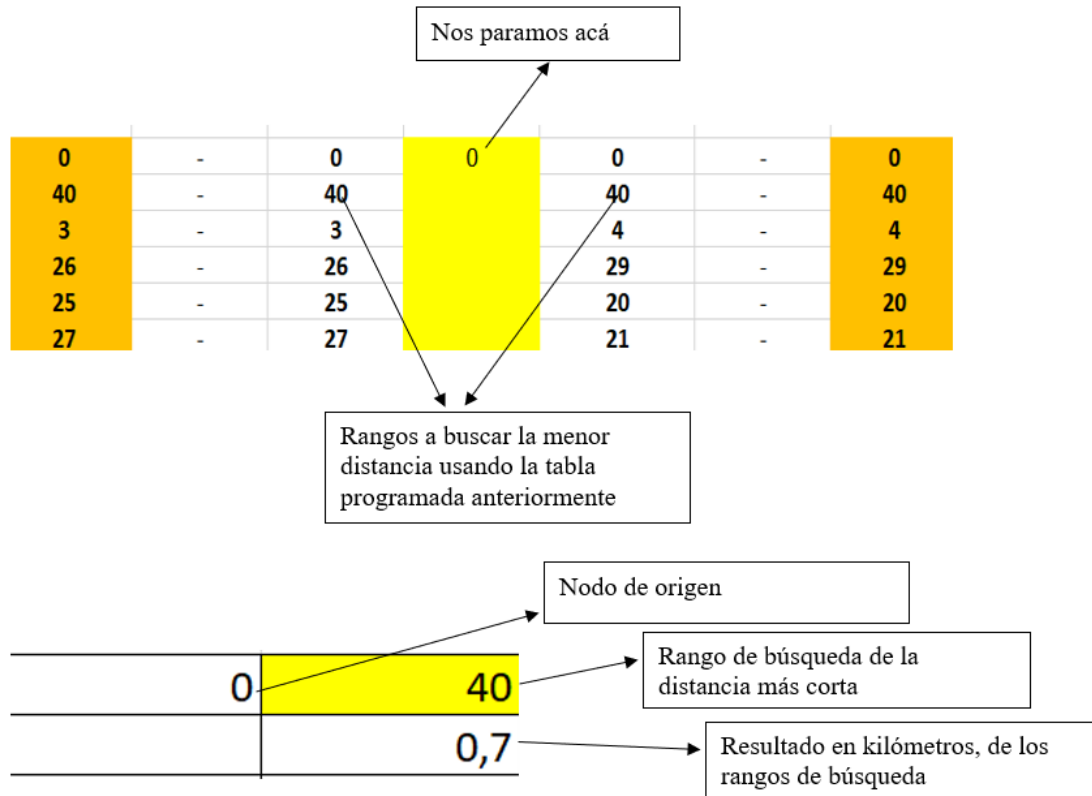
**Figura 22**

Esta la programacion de la celda de busqueda que debe de dejar fijo la celda donde esta el nodo 32 para poder arrastrar la formula y la celdas de las matrices.



**Figura 23**

**Paso4:** Una vez se tiene programada la tabla buscadora, se procederá a ubicarse en el nodo 0 y buscar la menor distancia (usando la tabla programada anteriormente) entre los rangos de 40 hasta 40 así:



**Figura 24**

Como la menor distancia entre los rangos desde 40 hasta 40 es el nodo 40 (con 0.7 kilómetros de distancia), partiendo desde el nodo cero, se procederá a colocarlo y se vuelve a realizar el mismo procedimiento partiendo ahora desde el nodo 40:

0	-	0	0	0	-	0
40	-	40	40	40	-	40
3	-	3		4	-	4
26	-	26		29	-	29
25	-	25		20	-	20

Ahora nos paramos acá

Repetimos el paso anterior que es buscar la distancia o el tiempo menor, usando la tabla programada anteriormente entre los rangos de 3 hasta 4 partiendo ahora desde el nodo 40 así:

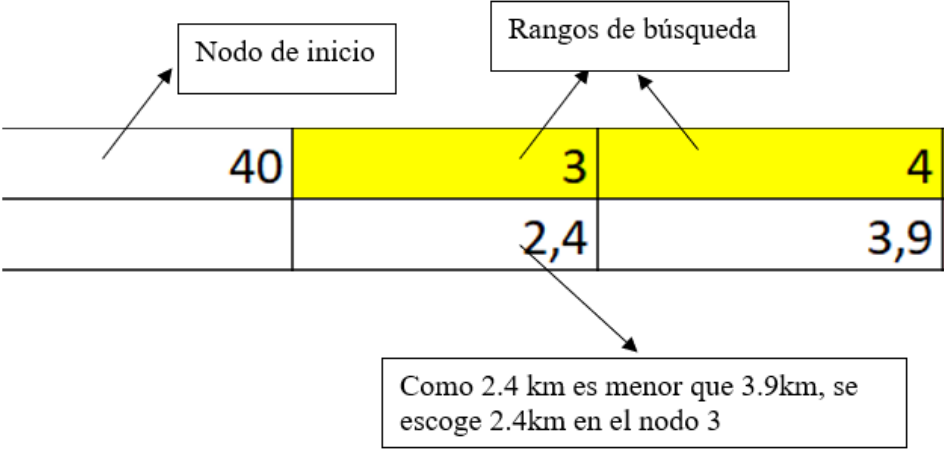


Figura 25

Como la menor distancia entre los rangos de 3 hasta 40 es 2,4 km, se escoge el nodo 3 y se coloca después del nodo 40 así:

0	-	0	0	0	-	0
40	-	40	40	40	-	40
3	-	3	3	4	-	4
26	-	26		29	-	29
25	-	25		20	-	20
27	-	27		21	-	21
24	-	24		22	-	22

Como ganó el nodo 3 con 2.4 km, se coloca después del nodo 40

Paso 5: Se repite el paso 4 así:

0	-	0	0	0	-	0
40	-	40	40	40	-	40
3	-	3	3	4	-	4
26	-	26		29	-	29
25	-	25		20	-	20
27	-	27		21	-	21
24	-	24		22	-	22

Nos paramos acá

	3	26	27	28	29
		0,23	3,1	4,8	3,6

Nodo de partida

Gana el nodo 26 con 0.23 km

Figura 26

Como gano el nodo 26 con 0.23 km, lo colocamos debajo del nodo 40 así:

0	-	0	0	0	-	0
40	-	40	40	40	-	40
3	-	3	3	4	-	4
26	-	26	26	29	-	29
25	-	25		20	-	20
27	-	27		21	-	21
24	-	24		22	-	22
28	-	28		23	-	23
29	-	29		26	-	26
19	-	19		3	-	3
18	-	18		25	-	25
2	-	2		27	-	27
11	-	11		24	-	24
39	-	39		28	-	28
13	-	13		30	-	30
14	-	14		32	-	32
10	-	10		19	-	19
15	-	15		2	-	2
12	-	12		18	-	18
38	-	38		17	-	17
7	-	7		16	-	16

**Figura 27**

Se sigue procediendo de esta manera hasta completar toda la columna central.

Una vez obtenida la ruta en la metodología de mutación se procederá a insertar y ejecutar en la macro de búsqueda tabú.

#### 8.14. Evitar repetidos en la mutación:

**Paso 1:** Se señala la columna central

**Paso 2:** Nos dirigimos a inicio, formato condicional, reglas para resaltar celdas

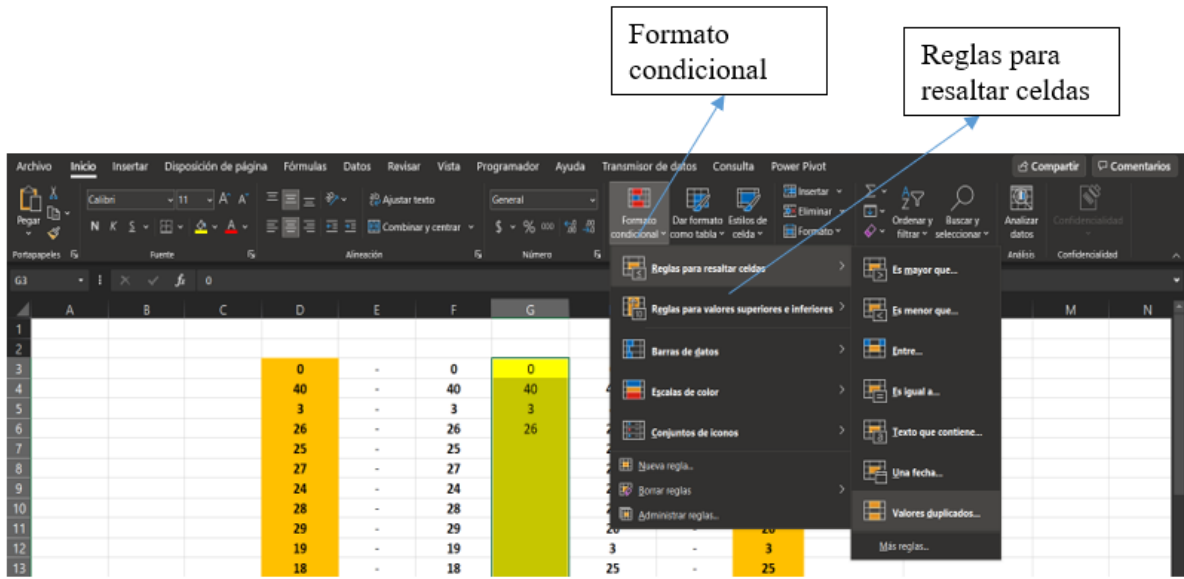


Figura 28

Se les deberá aparecer un letrero así:

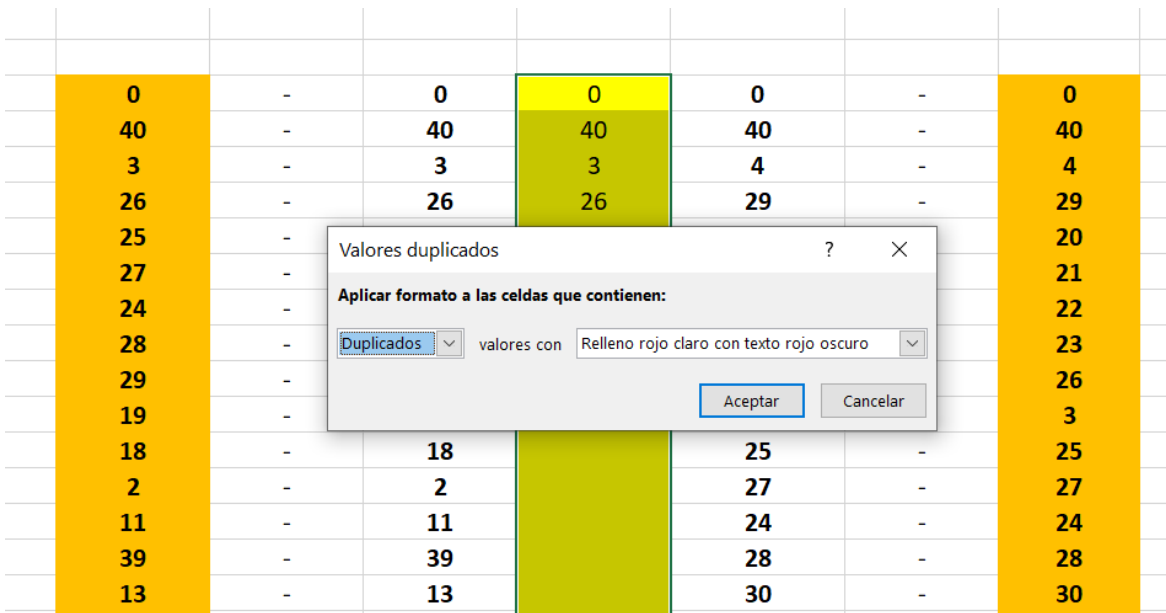


Figura 29

Damos clic en aceptar y automáticamente se nos marcara las celdas repetidas así:

0	-	0	0	0	-	0
40	-	40	40	40	-	40
3	-	3	3	4	-	4
26	-	26	26	29	-	29
25	-	25	26	20	-	20
27	-	27		21	-	21
24	-	24		22	-	22
28	-	28		23	-	23
29	-	29		26	-	26
10		10		2		2

Figura 30

Con este método se evita colocar nodos repetidos durante la realización de la mutación.

8.15. **Caso con nodo ocupado en la mutación:**

En caso de presentarse el nodo utilizado en la mutación así:

36	-	36	36	14	-	14
4	-	4	4	13	-	13
20	-	20		31	-	31
21	-	21		5	-	5
22	-	22		34	-	34
23	-	23		33	-	33
17	-	17		6	-	6
16	-	16		36	-	36
37	-	37		35	-	35
35	-	35		37	-	37
0	-	0		0	-	0

Figura 31

4	20	21	22	23	24	25	26	27	28	29	30	31
	2,2	3,8	4,9	4,6	5	2,6	3,2	4,1	3,9	1,8	7,6	6,8

↓  
Mínimo

Figura 32

Para este caso utilizando el método de mutación se obtiene que partiendo desde el nodo de origen 4, se busca entre los rangos la menor distancia en este caso, el cual arroja que el nodo 29 es el ganador y se procede a colocar debajo del 4, así:

36	-	36	36	14	-	14
4	-	4	4	13	-	13
20	-	20	29	31	-	31
21	-	21		5	-	5
22	-	22		34	-	34
23	-	23		33	-	33
17	-	17		6	-	6
16	-	16		36	-	36
37	-	37		35	-	35
35	-	35		37	-	37
0	-	0		0	-	0

**Figura 33**

Como denota la imagen anterior, el nodo 29 esta repetido, para este caso descartamos el nodo 29 y se procede a buscar el segundo nodo mejor así:

4	20	21	22	23	24	25	26	27	28	<del>29</del>	30	31
	2,2	3,8	4,9	4,6	5	2,6	3,2	4,1	3,9	<del>1,8</del>	7,6	6,8

2do Mínimo

**Figura 34**

Se coloca el nuevo nodo ganador y verificamos si realmente se puede usar, en caso de que tampoco se pueda usar, saltamos al tercer nodo y así sucesivamente hasta encontrar el mejor resultado.

5	-	5	5	12	-	12
33	-	33	33	11	-	11
6	-	6	6	39	-	39
36	-	36	36	14	-	14
4	-	4	4	13	-	13
20	-	20	20	31	-	31
21	-	21		5	-	5
22	-	22		34	-	34
23	-	23		33	-	33
17	-	17		6	-	6
16	-	16		36	-	36
37	-	37		35	-	35
35	-	35		37	-	37
0	-	0		0	-	0

**Figura 35**

Como se denota en la imagen anterior, el nodo 20 no está utilizado, por lo tanto, se puede emplear.

## **9. HIPOTESIS**

Como hipótesis se establecen las siguientes condiciones para el algoritmo de mutación:

### **9.1. Hipótesis nula**

La mutación permite mejorar o igualar la ruta inicial factible, generada por el algoritmo de búsqueda tabú clásica

### **9.2. Hipótesis alternativa**

La mutación no permite mejorar o igualar la ruta inicial factible, generada por el algoritmo de búsqueda tabú clásica

## 10. RESULTADOS:

La siguiente tabla muestra en resumen los resultados de obtenidos durante el desarrollo de la presente investigación:

### 10.1. Modelo 1 (Distancias):

	Ruta inicial factible	Búsqueda tabú	N° de iteraciones	Ranking
<b>Sin mutación</b>	190,33	184,23	5	2
<b>Con mutación</b>	187,33	184,23	3	1

Fuente: Elaboración propia

**Tabla 8**

Mejor ruta distancias:

0-40-26-3-25-27-24-28-29-19-18-2-11-39-13-14-12-10-15-38-7-9-1-8-31-30-32-34-5-33-6-36-4-20-21-23-22-16-17-37-35-0  
=184.23km

### 10.2. Modelo 2 (Tiempos)

	Ruta inicial factible	Búsqueda tabú	N° de iteraciones	Ranking
<b>Sin mutación</b>	487,3	446	7	1
<b>Con mutación</b>	487,3	446	7	1

Fuente: Elaboración propia

**Tabla 9**

Mejor ruta tiempos:

0-40-4-29-20-21-22-23-26-3-25-27-24-28-30-32-19-2-18-17-16-1-9-7-38-8-15-10-12-13-14-39-11-31-34-5-33-36-35-6-37-0  
=446 min

## 11. ANALISIS DE COSTOS:

El costo por kilómetros recorrido no contempla el costo de camiones que requieren cadena de frío.

### 11.1. Modelo 1 (Distancias)

MIN	KM	COSTO POR KM	TOTAL
531	190,33	\$ 11.821,13	\$ 2.249.916
518	184,23	\$ 11.821,13	\$ 2.177.806,78
524	187,33	\$ 11.821,13	\$ 2.214.452,28

Fuente: Elaboración propia

Tabla 10

### 11.2. Modelo 2 (Tiempos)

KM	COSTO POR KM	TIEMPO EN MIN	TOTAL
195,13	\$ 11.821,13	487,3	\$ 2.306.657,10
188,33	\$ 11.821,13	446	\$ 2.226.273,41

Fuente: Elaboración propia

Tabla 11

## 12. MEDICION DE LA CALIDAD:

Para medir la calidad del algoritmo se hace en base al número más corto de las iteraciones de cada una:

### 12.1. Calidad con respecto a modelo 1 (distancias):

Solución óptima: 3

Solución Metaheurística: 5

% calidad =  $(1 - (5-3) / 3) * 100 = 33,33\%$

### 12.2. Calidad con respecto a modelo 2 (tiempos):

Solución óptima: 7

Solución Metaheurística: 7

% calidad =  $(1 - (7-7) / 7) * 100 = 100\%$

### **13. RECOMENDACIONES DEL ALGORITMO DE MUTACION:**

Las siguientes son algunas recomendaciones con el fin de tener en cuenta a la hora de correr el algoritmo de mutación:

1. El procedimiento de mutación solo debe de hacer de arriba hacia abajo (Primero los nodos que se van a visitar al inicio de la ruta), esto con el fin de evitar seleccionar nodos inferiores que después deja sin elección en los rangos de la parte superior.
2. Se recomienda realizar dos corridas del algoritmo de mutación por matriz, esto con el fin de evitar errores de tipo humano que no permitan su correcta ejecución. Como trabajos futuros se recomienda la programación del algoritmo en un lenguaje de programación robusto.
3. La regla de paro del algoritmo está dada hasta volver a llegar de nuevo al cero (Punto de origen).

## 14. CONCLUSIONES

El uso de la adaptación de errores en los sistemas biológicos o mutaciones a métodos de resolución metaheurísticos, abre las puertas a múltiples posibilidades de mejora para los métodos clásicos, así como la invención de nuevos operadores de búsqueda, ya que, en la realización de la presente investigación, la búsqueda de artículos relacionados resultaron ser muy pocas, esto posiblemente a que muchos de los investigadores no manejen áreas de biología e ingeniería a la vez, lo que, quizás, conlleva a la limitación del diseño de metodologías nuevas. El conocimiento de una disciplina específica puede ser ajustada y generar resultados adecuados en otros campos del conocimiento.

La logística de distribución y control de temperatura de las vacunas COVID19 requiere de la aplicación de métodos metaheurísticos que en un corto tiempo de procesamiento de datos generen rutas eficientes en términos de tiempo y distancia recorrida.

Se evidencia que el algoritmo clásico de Búsqueda Tabú genera rutas eficientes para definir la secuencia de reparto de las vacunas por la red hospitalaria de Antioquia, garantizando que lleguen oportunamente a cada centro de vacunación. Sin embargo, aplicar nuevos operadores de búsqueda puede conllevar a mejorar el tiempo total de procesamiento del algoritmo de búsqueda de una ruta efectiva.

De acuerdo con Keifer & Effenberger, (1967), que define las metaheurísticas como “un proceso iterativo que guía una heurística subordinada, combinando diferentes conceptos para explorar y explotar las características que pueda exhibir el espacio de búsqueda, se puede concluir que a menor número de iteraciones por corrida en el algoritmo de búsqueda tabú, se logra ahorrar un tiempo considerable en términos de procesamiento lo que también conlleva a considerar la adaptación del proyecto a más o menos hospitales garantizando una respuesta cercana con menos operaciones en un periodo apretado de duración que exige una contestación inmediata.

Según los resultados expuestos anteriormente, la metodología de mutación permite aceptar la hipótesis nula y rechazar la hipótesis alternativa planteadas en la investigación, lo que, por ende, define el alcance de este nuevo procedimiento en términos de posibles mejoras a la metaheurística clásica de búsqueda tabú.

Con respecto al análisis de costos, basándose en los resultados se puede concluir que la ruta de distancias (Modelo 1) es más económica, con un ahorro considerable en términos monetarios a diferencia de la ruta de minimización de tiempos (Modelo 2), que si bien, es más rápida, requiere recorrer una mayor cantidad de kilómetros y por ende un mayor costo. El tomador de decisiones puede seleccionar cualquiera de las dos rutas dependiendo del presupuesto disponible y de la urgencia con que se quiera realizar el reparto de vacunas COVID-19.

## 15. DISCUSIÓN Y TRABAJOS FUTUROS

La metodología de mutación, al ser una nueva propuesta para mejorar el algoritmo de resolución de este tipo de problemas complejos, evidencia baja cantidad de artículos científicos acordes al tema central de la presente investigación, por lo tanto, se basó principalmente de otros ámbitos que difieren en la ingeniería industrial como lo son las áreas de microbiología y la genética para desarrollo del proceso de adaptación de errores biológicos, con lo cual se logra comprobar el mejoramiento en términos de procesamiento de la metaheurística de búsqueda tabú así como sus sorprendentes alcances.

Para futuros trabajos en los que se puede adaptar la presente investigación, se recomienda automatizar la “metodología de mutación” con el fin de acelerar los resultados en caso de necesitar una respuesta concreta en una ventana de tiempo muy corta.

Actualmente la presente investigación se puede adaptar para trabajos futuros que requieran cubrir más distancias o tiempos, gracias a la metodología de mutación que posiblemente acorta los tiempos de procesamiento del algoritmo de búsqueda tabú, es posible obtener respuestas más rápidas, no necesariamente la ruta óptima, pero si una buena solución aplicable en relación con el tiempo de resolución.

Las futuras investigaciones pueden incluir otros factores no contemplados, tales como: Ventanas de tiempo en la recepción del material transportado por parte de los hospitales que reciben las vacunas, la capacidad de carga dependiendo del tipo de vehículo utilizado, la disponibilidad de una flota de vehículos, la incidencia del tráfico o la disponibilidad de las vías para realizar la logística de distribución y los métodos de cargue y descargue del material en el vehículo.

## 16. CUMPLIMIENTO DE LOS OBJETIVOS DEL PROYECTO

Objetivo general / Especifico	Evidencia de cumplimiento
Diseñar rutas de reparto de vacunas COVID-19 desde el almacén distrital a la red hospitalaria de Antioquia, utilizando una mejora en la estrategia del algoritmo clásico de Búsqueda Tabú	Se logra evidenciar el cumplimiento a través de los objetivos específicos del proyecto.
Realizar una revisión bibliográfica de las diferentes estrategias que se han utilizado para resolver problemas de ruteo, haciendo énfasis en aquellos artículos que utilizaron como estrategia central un algoritmo de búsqueda tabú.	En los numerales 7.1, y 8.10 de la investigación, donde encuentran los resultados de la elaboración de los diferentes artículos consultados para este trabajo.
Modelar matemáticamente el problema de ruteo y recopilar información de los parámetros relacionados con distancia y tiempo entre los diferentes hospitales de la Red de Antioquia	En los numerales 8.1, 8.2, 8.3, y 8.4 de la presente investigación, se encuentran la descripción matemática, así como los datos usados para el análisis de los modelos y de la presente investigación.
Ejecutar el algoritmo de Búsqueda Tabú clásico y agregar un nuevo operador que mejore la estrategia de búsqueda	En los numeral 8.12, y 8.13 de la presente investigación se encuentra el paso a paso de la metodología de mutación, así como el instructivo de la macro usada para correr el algoritmo de búsqueda tabú, como el respectivo código en VBA en el anexo 1.
Determinar las rutas que mejoren la distancia y el tiempo total de entrega de vacunas COVID19 en una muestra de la red hospitalaria de Antioquia	En los numerales 10.1 y 10.2 de la tabla de la presente investigación, se encuentran las rutas resultantes derivadas del análisis de las corridas del algoritmo búsqueda, así como su análisis de calidad frente al procedimiento clásico vs el nuevo operador (metodología de mutación).

**Tabla 12**

## 17. CRONOGRAMA

N°	Actividad	Semanas											
		1	2	3	4	5	6	7	8	9	10	11	12
1	Recopilación de datos		x										
2	Aplicación vecino más cercano			x									
3	Adaptación, aplicación y diseño de metodología de mutación				x								
4	Determinación de rutas finales					x							
5	Redacción final del documento						x	x	x	x	x	x	

Fuente: Elaboración propia

**Tabla 13**

## 18. BIBLIOGRAFÍA

- ¿Qué es la Logística de distribución? (s/f). Recuperado el 15 de mayo de 2021, de <https://www.ceupe.com/blog/que-es-la-logistica-de-distribucion.html>
- 15 conceptos básicos de logística y cadena de suministro. (s/f). Recuperado el 15 de mayo de 2021, de <https://www.beetrack.com/es/blog/conceptos-basicos-de-logistica-cadena-de-suministr>
- Anaya Fuentes, G. E., Hernández Gress, E. S., Seck Tuoh Mora, J. C., & Medina Marín, J. (2016). Solución al Problema de Secuenciación de Trabajos mediante el Problema del Agente Viajero. *RIAI - Revista Iberoamericana de Automatica e Informatica Industrial*, 13(4), 430–437. <https://doi.org/10.1016/j.riai.2016.07.003>
- Bellidolo, S. M. (2013). Resolución del problema mínimo de ruteo. *Journal of Chemical Information and Modeling*, 53(9), 1689–1699.
- Cepeda, G., Lucas, M. S., & Delgado, M. S. E. (s/f). *ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL CENTRO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA “Diseño E Implementación De Una Heurística Para el Problema De Ruteo Vehicular Con Recolección y Entrega De Mercadería ( Vrppd ). ” Resumen.*
- COVID-19 : En Cundinamarca se perdieron más de 7.000 vacunas por interrumpir cadena de frío | Regiones | W Radio Colombia. (s/f). Recuperado el 14 de mayo de 2021, de <https://www.wradio.com.co/noticias/regionales/en-cundinamarca-se-perdieron-mas-de-7000-vacunas-por-interrumpir-cadena-de-frio/20210513/nota/4134454.aspx>
- Cruz Chávez, M., Moreno Bernal, P., & Peralta Abarca, J. (2014). Aplicación de la teoría de la complejidad en optimización combinatoria. *Inventio, la génesis de la cultura universitaria en Morelos*, 0(20), 35–42.
- Delgado Betancur, L., A. Becerra, M., Duque Mejia, C., Peluffo Ordoñez, D., & Alvarez Uribe, K. (2019). *Optimización del transporte público urbano mediante algoritmos de búsqueda Tabú y PSO* : 69–81.
- Garrido Lopez, A. J., Molina Ruiz, H. D., Almazan Vega, G., & Ryes Gonzales, I. (2009). The relationship between the level of completeness of a strategic performance management system and perceived advantages and disadvantages. *International Journal of Operations and Production Management*, 29(12), 1242–1265. <https://doi.org/10.1108/01443570911005983>
- Glover, F., & Melian, B. (2003). *Búsqueda Tabú. January.*
- Ivorra, C. (s/f). *Optimización con LINGO*. 1–25.
- Johnson, D. S., & McGeoch, L. A. (2018). The traveling salesman problem: *Local Search in Combinatorial Optimization*, 215–310. <https://doi.org/10.2307/j.ctv346t9c.13>
- Keifer, G., & Effenberger, F. (1967). Investigación operativa. *Angewandte Chemie International Edition*, 6(11), 951–952.
- Klug, W. S., Cummings, M. R., Spencer, C. A., & Palladino, M. A. (2012). *Concepts of Genetics William S. Klug, Michael R. Cummings, Charlotte A. Spencer & Michael A. Palladino.*
- Magallanes Bernal, E. U., Soberantes Puga, J. H., & Montesino Guerra, J. A. (2021). *Algoritmo genético y algoritmo de sistema de hormigas aplicados al problema del agente*. 4–10.
- Mejia Moncayo, C. (2012). *Modelo para la definición del layout de una celda de manufactura a través de optimización.*
- Melendez Acosta, J. N. (2020). *Recocido simulado y el algoritmo de selección clonal*

- aplicados al problema del agente viajero Simulated Annealing and Clonal Selection Algorithm applied to the Traveling Salesman Problem. 149(8), 1211–1226.*
- Ministerio de Salud y protección social. (2018). *Guía para el monitoreo de temperaturas en el transporte de biológicos.*
- Ministerio de Salud y protección social. (2021a). *Decreto 109 de 2021. 28.*
- Ministerio de Salud y protección social. (2021b). *Vacunas seguirán en cuarentena hasta que Invima otorgue aval para su aplicación.*  
<https://www.minsalud.gov.co/Paginas/Vacunas-seguiran-en-cuarentena-hasta-que-Invima-otorgue-aval-para-su-aplicacion.aspx>
- Morales, Jose Luis, A. (2004). *Modelo De Ruteo Dinámico Para La Operación De Transporte De Valores. 1, 9.*
- Moreno Pérez, J. (2004). *Metaheurísticas: Conceptos y Propiedades. 67.*  
<http://www.tebadm.ulpgc.es/almacen/seminarios/MH Las Palmas 2.pdf>
- Mota Verdura, S. (2016). *LA RECURSIÓN EN LA CIENCIA COGNITIVA DEL LENGUAJE: UNA INVESTIGACIÓN FORMAL Y EXPERIMENTAL.*
- Navarro, H. (2013). *Logística En La Cadena De Frio Mayo 2013 Proexport. 1–35.*  
[http://www.colombiatrade.com.co/sites/default/files/conferencia\\_logistica\\_en\\_la\\_cadena\\_de\\_frio\\_proexport\\_2013.pdf](http://www.colombiatrade.com.co/sites/default/files/conferencia_logistica_en_la_cadena_de_frio_proexport_2013.pdf)
- Nodo Optimización de CPLEX - IBM Documentation.* (s/f). Recuperado el 10 de mayo de 2021, de <https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=nodes-cplex-optimization-node>
- Paul, J., & Dvm, C. (s/f). *Dinámica de infección e inmunidad de rebaño en poblaciones porcinas. Figura 1.*
- Real academia española. (s/f). *vacuno, vacuna | Definición | Diccionario de la lengua española | RAE - ASALE.* Recuperado el 15 de abril de 2021, de <https://dle.rae.es/vacuno>
- Roberto, C. P., Jonatan, G. P., & Leon, E. (2008). GRISLAS: Un algoritmo genético paralelo que combina los modelos de grillas e islas para encontrar soluciones óptimas cercanas al problema del agente viajero. *Revista Avances en Sistemas e Informática, 5(3), 13–19.*
- Ruta de transporte: tipos, diseño y planificación de rutas logísticas.* (s/f). Recuperado el 15 de mayo de 2021, de <https://www.beetrack.com/es/blog/ruta-de-transporte-diseñarla>
- S.Hillier, F., & J. Lieberman, G. (2010). *Investigacion de operaciones.*
- Salud, P. D. E., Entidades, I. P. S. Y., Contra, D. E. V., & Covid, E. L. (2021). *Radicado:K 2021090000042 Fecha: 08/02/2021. 383 5608.*
- Torres Cepeda, C. (2021). *Vacunas covid en Tolima: en cuarentena por alteración de cadena de frío | La FM.* <https://www.lafm.com.co/colombia/ponen-en-cuarentena-mas-de-5700-vacunas-anticovid-que-llegaron-al-tolima>

## 19. ANEXOS:

### Anexo 1.

Sub primero\_intento\_3()

'Definir variables, una por cada cliente a visitar

Dim a As Integer

Dim b As Integer

Dim c As Integer

Dim d As Integer

Dim e As Integer

Dim f As Integer

Dim g As Integer

Dim h As Integer

Dim k As Integer

Dim l As Integer

Dim m As Integer

Dim n As Integer

Dim o As Integer

Dim p As Integer

Dim q As Integer

Dim r As Integer

Dim s As Integer

Dim t As Integer

Dim u As Integer

Dim v As Integer

Dim w As Integer

Dim x As Integer

Dim y As Integer

Dim z As Integer

Dim aa As Integer

Dim ab As Integer

Dim ac As Integer

Dim ad As Integer

Dim ae As Integer

Dim af As Integer

Dim ag As Integer

Dim ah As Integer

Dim ak As Integer

Dim al As Integer

Dim am As Integer

Dim an As Integer

Dim ao As Integer

Dim ap As Integer

Dim aq As Integer

Dim ar As Integer

'Solicitar ruta inicial

```
a = Val(InputBox("Primer Cliente a visitar?", , 1))
b = Val(InputBox("Segundo Cliente a visitar?", , 2))
c = Val(InputBox("Tercer Cliente a visitar?", , 3))
d = Val(InputBox("Cuarto Cliente a visitar?", , 4))
e = Val(InputBox("Quinto Cliente a visitar?", , 5))
f = Val(InputBox("Sexto Cliente a visitar?", , 6))
g = Val(InputBox("Séptimo Cliente a visitar?", , 7))
h = Val(InputBox("Octavo Cliente a visitar?", , 8))
k = Val(InputBox("Noveno Cliente a visitar?", , 9))
l = Val(InputBox("Décimo Cliente a visitar?", , 10))
m = Val(InputBox("Undécimo Cliente a visitar?", , 11))
n = Val(InputBox("Duodécimo Cliente a visitar?", , 12))
o = Val(InputBox("Decimotercer Cliente a visitar?", , 13))
p = Val(InputBox("Decimocuarto Cliente a visitar?", , 14))
q = Val(InputBox("Decimoquinto Cliente a visitar?", , 15))
r = Val(InputBox("Decimosexto Cliente a visitar?", , 16))
s = Val(InputBox("Decimoseptimo Cliente a visitar?", , 17))
t = Val(InputBox("Decimooctavo Cliente a visitar?", , 18))
u = Val(InputBox("Decimonoveno Cliente a visitar?", , 19))
v = Val(InputBox("Veintiabo Cliente a visitar?", , 20))
w = Val(InputBox("Veintiun Cliente a visitar?", , 21))
x = Val(InputBox("veintidos Cliente a visitar?", , 22))
y = Val(InputBox("veintitres Cliente a visitar?", , 23))
z = Val(InputBox("veinticuatro Cliente a visitar?", , 24))
aa = Val(InputBox("veinticinco Cliente a visitar?", , 25))
ab = Val(InputBox("veinticeis Cliente a visitar?", , 26))
ac = Val(InputBox("veinticiete Cliente a visitar?", , 27))
ad = Val(InputBox("veintiocho Cliente a visitar?", , 28))
ae = Val(InputBox("veintinueve Cliente a visitar?", , 29))
af = Val(InputBox("Treinta Cliente a visitar?", , 30))
ag = Val(InputBox("Treintayuno Cliente a visitar?", , 31))
ah = Val(InputBox("Treintaydos Cliente a visitar?", , 32))
ak = Val(InputBox("Treintaytres Cliente a visitar?", , 33))
al = Val(InputBox("Treintaycuatro Cliente a visitar?", , 34))
am = Val(InputBox("Treintaycinco Cliente a visitar?", , 35))
an = Val(InputBox("Treintayceis Cliente a visitar?", , 36))
ao = Val(InputBox("Trintaysiete Cliente a visitar?", , 37))
ap = Val(InputBox("Trintayocho Cliente a visitar?", , 38))
aq = Val(InputBox("Trintaynueve Cliente a visitar?", , 39))
ar = Val(InputBox("Cuarenta Cliente a visitar?", , 40))
```

'Colocar texto del Subviaje Inverso

```
Range("A44").Select  
Selection.Font.Bold = True  
Range("A44") = "Solución a Analizar con el Heurístico de subviaje inverso"
```

'Llenar tabla con 150 veces la ruta inicial

Dim i As Integer

Dim j As Integer

For i = 1 To 150

Hoja1.Cells(i + 49, 1) = "0"

Hoja1.Cells(i + 49, 2) = a

Hoja1.Cells(i + 49, 3) = b

Hoja1.Cells(i + 49, 4) = c

Hoja1.Cells(i + 49, 5) = d

Hoja1.Cells(i + 49, 6) = e

Hoja1.Cells(i + 49, 7) = f

Hoja1.Cells(i + 49, 8) = g

Hoja1.Cells(i + 49, 9) = h

Hoja1.Cells(i + 49, 10) = k

Hoja1.Cells(i + 49, 11) = l

Hoja1.Cells(i + 49, 12) = m

Hoja1.Cells(i + 49, 13) = n

Hoja1.Cells(i + 49, 14) = o

Hoja1.Cells(i + 49, 15) = p

Hoja1.Cells(i + 49, 16) = q

Hoja1.Cells(i + 49, 17) = r

Hoja1.Cells(i + 49, 18) = s

Hoja1.Cells(i + 49, 19) = t

Hoja1.Cells(i + 49, 20) = u

Hoja1.Cells(i + 49, 21) = v

Hoja1.Cells(i + 49, 22) = w

Hoja1.Cells(i + 49, 23) = x

Hoja1.Cells(i + 49, 24) = y

Hoja1.Cells(i + 49, 25) = z

Hoja1.Cells(i + 49, 26) = aa

Hoja1.Cells(i + 49, 27) = ab

Hoja1.Cells(i + 49, 28) = ac

Hoja1.Cells(i + 49, 29) = ad

Hoja1.Cells(i + 49, 30) = ae

Hoja1.Cells(i + 49, 31) = af

Hoja1.Cells(i + 49, 32) = ag

Hoja1.Cells(i + 49, 33) = ah

Hoja1.Cells(i + 49, 34) = ak

Hoja1.Cells(i + 49, 35) = al

Hoja1.Cells(i + 49, 36) = am

Hoja1.Cells(i + 49, 37) = an

Hoja1.Cells(i + 49, 38) = ao

Hoja1.Cells(i + 49, 39) = ap  
Hoja1.Cells(i + 49, 40) = aq  
Hoja1.Cells(i + 49, 41) = ar  
Hoja1.Cells(i + 49, 42) = "0"

Next i

Heurístico del subviaje inverso para 2, 3, 4 y 5 posiciones

Hoja1.Cells(50, 2) = b  
Hoja1.Cells(50, 3) = a  
Hoja1.Cells(51, 3) = c  
Hoja1.Cells(51, 4) = b  
Hoja1.Cells(52, 4) = d  
Hoja1.Cells(52, 5) = c  
Hoja1.Cells(53, 5) = e  
Hoja1.Cells(53, 6) = d  
Hoja1.Cells(54, 6) = f  
Hoja1.Cells(54, 7) = e  
Hoja1.Cells(55, 7) = g  
Hoja1.Cells(55, 8) = f  
Hoja1.Cells(56, 8) = h  
Hoja1.Cells(56, 9) = g  
Hoja1.Cells(57, 9) = k  
Hoja1.Cells(57, 10) = h  
Hoja1.Cells(58, 10) = l  
Hoja1.Cells(58, 11) = k  
Hoja1.Cells(59, 11) = m  
Hoja1.Cells(59, 12) = l  
Hoja1.Cells(60, 12) = n  
Hoja1.Cells(60, 13) = m  
Hoja1.Cells(61, 13) = o  
Hoja1.Cells(61, 14) = n  
Hoja1.Cells(62, 14) = p  
Hoja1.Cells(62, 15) = o  
Hoja1.Cells(63, 15) = q  
Hoja1.Cells(63, 16) = p  
Hoja1.Cells(64, 16) = r  
Hoja1.Cells(64, 17) = q  
Hoja1.Cells(65, 17) = s  
Hoja1.Cells(65, 18) = r  
Hoja1.Cells(66, 18) = t  
Hoja1.Cells(66, 19) = s  
Hoja1.Cells(67, 19) = u  
Hoja1.Cells(67, 20) = t

Hoja1.Cells(68, 20) = v  
Hoja1.Cells(68, 21) = u  
Hoja1.Cells(69, 21) = w  
Hoja1.Cells(69, 22) = v  
Hoja1.Cells(70, 22) = x  
Hoja1.Cells(70, 23) = w  
Hoja1.Cells(71, 23) = y  
Hoja1.Cells(71, 24) = x  
Hoja1.Cells(72, 24) = z  
Hoja1.Cells(72, 25) = y  
Hoja1.Cells(73, 25) = aa  
Hoja1.Cells(73, 26) = z  
Hoja1.Cells(74, 26) = ab  
Hoja1.Cells(74, 27) = aa  
Hoja1.Cells(75, 27) = ac  
Hoja1.Cells(75, 28) = ab  
Hoja1.Cells(76, 28) = ad  
Hoja1.Cells(76, 29) = ac  
Hoja1.Cells(77, 29) = ae  
Hoja1.Cells(77, 30) = ad  
Hoja1.Cells(78, 30) = af  
Hoja1.Cells(78, 31) = ae  
Hoja1.Cells(79, 31) = ag  
Hoja1.Cells(79, 32) = af  
Hoja1.Cells(80, 32) = ah  
Hoja1.Cells(80, 33) = ag  
Hoja1.Cells(81, 33) = ak  
Hoja1.Cells(81, 34) = ah  
Hoja1.Cells(82, 34) = al  
Hoja1.Cells(82, 35) = ak  
Hoja1.Cells(83, 35) = am  
Hoja1.Cells(83, 36) = al  
Hoja1.Cells(84, 36) = an  
Hoja1.Cells(84, 37) = am  
Hoja1.Cells(85, 37) = ao  
Hoja1.Cells(85, 38) = an  
Hoja1.Cells(86, 38) = ap  
Hoja1.Cells(86, 39) = ao  
Hoja1.Cells(87, 39) = aq  
Hoja1.Cells(87, 40) = ap  
Hoja1.Cells(88, 40) = ar  
Hoja1.Cells(88, 41) = aq  
Hoja1.Cells(89, 2) = c  
Hoja1.Cells(89, 4) = a  
Hoja1.Cells(90, 3) = d  
Hoja1.Cells(90, 5) = b  
Hoja1.Cells(91, 4) = e

Hoja1.Cells(91, 6) = c  
Hoja1.Cells(92, 5) = f  
Hoja1.Cells(92, 7) = d  
Hoja1.Cells(93, 6) = g  
Hoja1.Cells(93, 8) = e  
Hoja1.Cells(94, 7) = h  
Hoja1.Cells(94, 9) = f  
Hoja1.Cells(95, 8) = k  
Hoja1.Cells(95, 10) = g  
Hoja1.Cells(96, 9) = l  
Hoja1.Cells(96, 11) = h  
Hoja1.Cells(97, 10) = m  
Hoja1.Cells(97, 12) = k  
Hoja1.Cells(98, 11) = n  
Hoja1.Cells(98, 13) = l  
Hoja1.Cells(99, 12) = o  
Hoja1.Cells(99, 14) = m  
Hoja1.Cells(100, 13) = p  
Hoja1.Cells(100, 15) = n  
Hoja1.Cells(101, 14) = q  
Hoja1.Cells(101, 16) = o  
Hoja1.Cells(102, 15) = r  
Hoja1.Cells(102, 17) = p  
Hoja1.Cells(103, 16) = s  
Hoja1.Cells(103, 18) = q  
Hoja1.Cells(104, 17) = t  
Hoja1.Cells(104, 19) = r  
Hoja1.Cells(105, 18) = u  
Hoja1.Cells(105, 20) = s  
Hoja1.Cells(106, 19) = v  
Hoja1.Cells(106, 21) = t  
Hoja1.Cells(107, 20) = w  
Hoja1.Cells(107, 22) = u  
Hoja1.Cells(108, 21) = x  
Hoja1.Cells(108, 23) = v  
Hoja1.Cells(109, 22) = y  
Hoja1.Cells(109, 24) = w  
Hoja1.Cells(110, 23) = z  
Hoja1.Cells(110, 25) = x  
Hoja1.Cells(111, 24) = aa  
Hoja1.Cells(111, 26) = y  
Hoja1.Cells(112, 25) = ab  
Hoja1.Cells(112, 27) = z  
Hoja1.Cells(113, 26) = ac  
Hoja1.Cells(113, 28) = aa  
Hoja1.Cells(114, 27) = ad  
Hoja1.Cells(114, 29) = ab

Hoja1.Cells(115, 28) = ae  
Hoja1.Cells(115, 30) = ac  
Hoja1.Cells(116, 29) = af  
Hoja1.Cells(116, 31) = ad  
Hoja1.Cells(117, 30) = ag  
Hoja1.Cells(117, 32) = ae  
Hoja1.Cells(118, 31) = ah  
Hoja1.Cells(118, 33) = af  
Hoja1.Cells(119, 32) = ak  
Hoja1.Cells(119, 34) = ag  
Hoja1.Cells(120, 33) = al  
Hoja1.Cells(120, 35) = ah  
Hoja1.Cells(121, 34) = am  
Hoja1.Cells(121, 36) = ak  
Hoja1.Cells(122, 35) = an  
Hoja1.Cells(122, 37) = al  
Hoja1.Cells(123, 36) = ao  
Hoja1.Cells(123, 38) = am  
Hoja1.Cells(124, 37) = ap  
Hoja1.Cells(124, 39) = an  
Hoja1.Cells(125, 38) = aq  
Hoja1.Cells(125, 40) = ao  
Hoja1.Cells(126, 39) = ar  
Hoja1.Cells(126, 41) = ap  
Hoja1.Cells(127, 2) = d  
Hoja1.Cells(127, 5) = a  
Hoja1.Cells(128, 3) = e  
Hoja1.Cells(128, 6) = b  
Hoja1.Cells(129, 4) = f  
Hoja1.Cells(129, 7) = c  
Hoja1.Cells(130, 5) = g  
Hoja1.Cells(130, 8) = d  
Hoja1.Cells(131, 6) = h  
Hoja1.Cells(131, 9) = e  
Hoja1.Cells(132, 7) = k  
Hoja1.Cells(132, 10) = f  
Hoja1.Cells(133, 8) = l  
Hoja1.Cells(133, 11) = g  
Hoja1.Cells(134, 9) = m  
Hoja1.Cells(134, 12) = h  
Hoja1.Cells(135, 10) = n  
Hoja1.Cells(135, 13) = k  
Hoja1.Cells(136, 11) = o  
Hoja1.Cells(136, 14) = l  
Hoja1.Cells(137, 12) = p  
Hoja1.Cells(137, 15) = m  
Hoja1.Cells(138, 13) = q

Hoja1.Cells(138, 16) = n  
Hoja1.Cells(139, 14) = r  
Hoja1.Cells(139, 17) = o  
Hoja1.Cells(140, 15) = s  
Hoja1.Cells(140, 18) = p  
Hoja1.Cells(141, 16) = t  
Hoja1.Cells(141, 19) = q  
Hoja1.Cells(142, 17) = u  
Hoja1.Cells(142, 20) = r  
Hoja1.Cells(143, 18) = v  
Hoja1.Cells(143, 21) = s  
Hoja1.Cells(144, 19) = w  
Hoja1.Cells(144, 22) = t  
Hoja1.Cells(145, 20) = x  
Hoja1.Cells(145, 23) = u  
Hoja1.Cells(146, 21) = y  
Hoja1.Cells(146, 24) = v  
Hoja1.Cells(147, 22) = z  
Hoja1.Cells(147, 25) = w  
Hoja1.Cells(148, 23) = aa  
Hoja1.Cells(148, 26) = x  
Hoja1.Cells(149, 24) = ab  
Hoja1.Cells(149, 27) = y  
Hoja1.Cells(150, 25) = ac  
Hoja1.Cells(150, 28) = z  
Hoja1.Cells(151, 26) = ad  
Hoja1.Cells(151, 29) = aa  
Hoja1.Cells(152, 27) = ae  
Hoja1.Cells(152, 30) = ab  
Hoja1.Cells(153, 28) = af  
Hoja1.Cells(153, 31) = ac  
Hoja1.Cells(154, 29) = ag  
Hoja1.Cells(154, 32) = ad  
Hoja1.Cells(155, 30) = ah  
Hoja1.Cells(155, 33) = ae  
Hoja1.Cells(156, 31) = ak  
Hoja1.Cells(156, 34) = af  
Hoja1.Cells(157, 32) = al  
Hoja1.Cells(157, 35) = ag  
Hoja1.Cells(158, 33) = am  
Hoja1.Cells(158, 36) = ah  
Hoja1.Cells(159, 34) = an  
Hoja1.Cells(159, 37) = ak  
Hoja1.Cells(160, 35) = ao  
Hoja1.Cells(160, 38) = al  
Hoja1.Cells(161, 36) = ap  
Hoja1.Cells(161, 39) = am

Hoja1.Cells(162, 37) = aq  
Hoja1.Cells(162, 40) = an  
Hoja1.Cells(163, 41) = ao  
Hoja1.Cells(163, 38) = ar  
Hoja1.Cells(164, 2) = e  
Hoja1.Cells(164, 6) = a  
Hoja1.Cells(165, 3) = f  
Hoja1.Cells(165, 7) = b  
Hoja1.Cells(166, 4) = g  
Hoja1.Cells(166, 8) = c  
Hoja1.Cells(167, 5) = h  
Hoja1.Cells(167, 9) = d  
Hoja1.Cells(168, 6) = k  
Hoja1.Cells(168, 10) = e  
Hoja1.Cells(169, 7) = l  
Hoja1.Cells(169, 11) = f  
Hoja1.Cells(170, 8) = m  
Hoja1.Cells(170, 12) = g  
Hoja1.Cells(171, 9) = n  
Hoja1.Cells(171, 13) = h  
Hoja1.Cells(172, 10) = o  
Hoja1.Cells(172, 14) = k  
Hoja1.Cells(173, 11) = p  
Hoja1.Cells(173, 15) = l  
Hoja1.Cells(174, 12) = q  
Hoja1.Cells(174, 16) = m  
Hoja1.Cells(175, 13) = r  
Hoja1.Cells(175, 17) = n  
Hoja1.Cells(176, 14) = s  
Hoja1.Cells(176, 18) = o  
Hoja1.Cells(177, 15) = t  
Hoja1.Cells(177, 19) = p  
Hoja1.Cells(178, 16) = u  
Hoja1.Cells(178, 20) = q  
Hoja1.Cells(179, 17) = v  
Hoja1.Cells(179, 21) = r  
Hoja1.Cells(180, 18) = w  
Hoja1.Cells(180, 22) = s  
Hoja1.Cells(181, 19) = x  
Hoja1.Cells(181, 23) = t  
Hoja1.Cells(182, 20) = y  
Hoja1.Cells(182, 24) = u  
Hoja1.Cells(183, 21) = z  
Hoja1.Cells(183, 25) = v  
Hoja1.Cells(184, 22) = aa  
Hoja1.Cells(184, 26) = w  
Hoja1.Cells(185, 23) = ab

Hoja1.Cells(185, 27) = x  
Hoja1.Cells(186, 24) = ac  
Hoja1.Cells(186, 28) = y  
Hoja1.Cells(187, 25) = ad  
Hoja1.Cells(187, 29) = z  
Hoja1.Cells(188, 26) = ae  
Hoja1.Cells(188, 30) = aa  
Hoja1.Cells(189, 27) = af  
Hoja1.Cells(189, 31) = ab  
Hoja1.Cells(190, 28) = ag  
Hoja1.Cells(190, 32) = ac  
Hoja1.Cells(191, 29) = ah  
Hoja1.Cells(191, 33) = ad  
Hoja1.Cells(192, 30) = ak  
Hoja1.Cells(192, 34) = ae  
Hoja1.Cells(193, 31) = al  
Hoja1.Cells(193, 35) = af  
Hoja1.Cells(194, 32) = am  
Hoja1.Cells(194, 36) = ag  
Hoja1.Cells(195, 33) = an  
Hoja1.Cells(195, 37) = ah  
Hoja1.Cells(196, 34) = ao  
Hoja1.Cells(196, 38) = ak  
Hoja1.Cells(197, 35) = ap  
Hoja1.Cells(197, 39) = al  
Hoja1.Cells(198, 36) = aq  
Hoja1.Cells(198, 40) = am  
Hoja1.Cells(199, 37) = ar  
Hoja1.Cells(199, 41) = an

'Calcular Valor de la función objetivo de cada ruta

For i = 1 To 150

For j = 1 To 42

Hoja1.Cells(45, j) = Hoja1.Cells(i + 49, j)

Next j

Hoja1.Cells(i + 49, 44) = Hoja1.Cells(45, 44)

Next i

'Dejar Ruta Inicial como ruta original de partida

Range("A45") = "0"

Range("B45") = a

Range("C45") = b

Range("D45") = c

Range("E45") = d

Range("F45") = e

Range("G45") = f

Range("H45") = g

Range("I45") = h  
Range("J45") = k  
Range("K45") = l  
Range("L45") = m  
Range("M45") = n  
Range("N45") = o  
Range("O45") = p  
Range("P45") = q  
Range("Q45") = r  
Range("R45") = s  
Range("S45") = t  
Range("T45") = u  
Range("U45") = v  
Range("V45") = w  
Range("W45") = x  
Range("X45") = y  
Range("Y45") = z  
Range("Z45") = aa  
Range("AA45") = ab  
Range("AB45") = ac  
Range("AC45") = ad  
Range("AD45") = ae  
Range("AE45") = af  
Range("AF45") = ag  
Range("AG45") = ah  
Range("AH45") = ak  
Range("AI45") = al  
Range("AJ45") = am  
Range("AK45") = an  
Range("AL45") = ao  
Range("AM45") = ap  
Range("AN45") = aq  
Range("AO45") = ar  
Range("Ap45") = "0"

End Sub