Universidad Nacional Mayor de San Marcos
Universidad del Perú. Decana de América
Facultad de Ciencias Matemáticas
Escuela Profesional de Investigación Operativa

Propuesta de rutas para patrullaje mediante metaheurísticas para optimizar el recorrido de puntos críticos en el distrito del Rímac

TESIS

Para optar el Título Profesional de Licenciado en Investigación Operativa

AUTOR

Alejandro Fabio Salvatore AYLAS BARRANCA

ASESOR

Mg. Mario Edison NINAQUISPE SOTO

Lima, Perú

2022



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

https://creativecommons.org/licenses/by-nc-sa/4.0/

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Aylas, A. (2022). *Propuesta de rutas para patrullaje mediante metaheurísticas para optimizar el recorrido de puntos críticos en el distrito del Rímac*. [Tesis de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ciencias Matemáticas, Escuela Profesional de Investigación Operativa]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Alejandro Fabio Salvatore Aylas Barranca
Tipo de documento de identidad	DNI
Número de documento de identidad	70045020
Datos de asesor	
Nombres y apellidos	Mario Edison Ninaquispe Soto
Tipo de documento de identidad	DNI
Número de documento de identidad	41887115
URL de ORCID	https://orcid.org/0000-0002-6287-3291
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Esther Berger Vidal
Tipo de documento	DNI
Número de documento de identidad	08766040
Miembro del jurado 1	
Nombres y apellidos	José Carlos Oré Luján
Tipo de documento	DNI
Número de documento de identidad	06119405
Datos de investigación	
Línea de investigación	A.3.3.1. Optimización Matemática
Grupo de investigación	No aplica
Agencia de financiamiento	Sin financiamiento.

Ubicación geográfica de la investigación	Universidad Nacional Mayor de San Marcos País: Perú Departamento: Lima Provincia: Lima Distrito: Lima Coordenadas geográficas Latitud: -12.058333 Longitud: -77.083333
Año o rango de años en que se realizó la investigación	Noviembre 2021
URL de disciplinas OCDE	Matemáticas aplicadas https://purl.org/pe-repo/ocde/ford#1.01.02

Anexo 6

ACTA DE SUSTENTACIÓN DE TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE LICENCIADO EN INVESTIGACIÓN OPERATIVA

En la Ciudad Universitaria, Facultad de Ciencias Matemáticas, siendo las 18:15 horas del día viernes 4 de de febrero del año 2022, se reunieron los docentes designados como Miembros del Jurado Evaluador de Tesis:

Dra. Esther Berger Vidal	(Presidenta)
Mg. José Carlos Oré Luján	(Miembro)
Mg. Mario Edison Ninaquispe Soto	(Miembro-Asesor)

Para la sustentación de la Tesis intitulada: "PROPUESTA DE RUTAS PARA PATRULLAJE MEDIANTE METAHEURÍSTICAS PARA OPTIMIZAR EL RECORRIDO DE PUNTOS CRÍTICOS EN EL DISTRITO DEL RÍMAC", presentada por el señor Bachiller Alejandro Fabio Salvatore Aylas Barranca, para obtener el Título Profesional de Licenciado en Investigación Operativa.

Luego de la exposición de la Tesis, la Presidenta invitó al expositor a dar respuesta a las preguntas formuladas.

Realizada la evaluación correspondiente por los miembros del jurado, el expositor mereció la aprobación APROBADO CON MENCIÓN HONROSA, con un calificativo promedio de DIECIOCHO (18).

A continuación, los miembros del jurado dan manifiesto que el participante señor Bachiller Alejandro Fabio Salvatore Aylas Barranca, en virtud de haber aprobado la sustentación de su tesis, será propuesto para que se le otorgue el TítuloProfesional de Licenciado en Investigación Operativa.

Siendo las 18:50 horas, se levantó la Sesión, firmando para constancia la presente Acta en tres (3) copias originales o archivo PDF.

PRESIDENTA
Dra. Esther Berger Vidal

MIEMBRO
Mg. José Carlos Oré Luján

MIEMBRO-ASESOR
Mg. Mario Edison Ninaquispe Soto

Esta es una copia auténtica imprimible de un documento electrónico archivado por la Universidad Nacional Mayor de San Marcos, aplicando lo dispuesto por el Art. 25 de D.S. 070-2013-PCM y la Tercera Disposición Complementaria Final del D.S. 026- 2016-PCM. Su autenticidad e integridad pueden ser contrastadas a través de lasiguiente dirección web: **https://spsgd.unmsm.edu.pe/verifica/inicio.do** e ingresando el siguiente código de verificación: **FVCRJUX**

Resumen

Propuesta de rutas para patrullaje mediante metaheurísticas para optimizar el recorrido de puntos críticos en el distrito del Rímac

Alejandro Fabio Salvatore Aylas Barranca

Enero 2022

Título obtenido : Licenciado en Investigación Operativa

La trayectoria de los vehículos policiales durante las rutinas de patrullaje debe incluir puntos críticos y lugares vulnerables a delitos; sin embargo, la elección de rutas en el distrito del Rímac se lleva a cabo de manera empírica, generando largos recorridos. Se planteó como objetivo principal el diseño de rutas para el distrito del Rímac mediante la aplicación de dos heurísticas y dos metaheurísticas, además con el lenguaje de programación en Visual Basic, se hizo un aplicativo desktop (o de escritorio) para computadora, este brindó varias alternativas de rutas cercanos al óptimo para patrullaje basadas en las soluciones de las técnicas matemáticas. Con la recolección de datos sobre la actividad del patrullaje vehicular de enero hasta junio del 2021 de todas las comisarías del distrito, se planteó un método para replicar los recorridos empíricos. Los resultados mostraron que utilizando las técnicas mencionadas, estas diseñan rutas con menor distancia en comparación a los recorridos empíricos, con la validación de las hipótesis se concluyó que usando heurísticas y metaheurísticas se logra minimizar la distancia de los recorridos empíricos, por lo que las rutas hechas por algoritmos matemáticos optimizan el recorrido de puntos críticos. Finalmente se presentaron las propuestas de rutas para las cinco comisarias del distrito del Rímac.

Palabras clave: Metaheurística, Heurística, Patrullaje vehicular, Recorrido, Puntos críticos

Abstract

Proposal of routes for patrolling with metaheuristics to optimize the route of hot spots in the district of Rimac

Alejandro Fabio Salvatore Aylas Barranca

January 2022

Degree obtained : Graduate in Operations Research

The trajectory of police vehicles during patrol routines must include hot spots and places vulnerable to crime; however, the choice of routes in the Rimac district is made empirically, generating long routes. The main objective was the design routes for the district of Rimac with the application of two heuristics and two metaheuristics, in addition with the programming language in Visual Basic, a desktop application for computer was made, it provided several route alternatives near to the optimum for patrol based on the solutions of the mathematical techniques. Information was gathered on vehicle patrol activities from January to June 2021 from all police stations in the district, a method was proposed to replicate the empirical routes. The results showed that using the techniques mentioned, they design routes with less distance compared to the empirical routes, with the validation of the hypotheses it was concluded that using heuristics and metaheuristics it is possible to minimize the distance of the empirical routes, so that the routes made by mathematical algorithms optimize the route of critical points. Finally, the route proposals for the five police stations of the Rimac district were presented.

Keywords: Metaheuristic, Heuristic, Vehicle patrolling, Routes, Hot Spots

Dedicatoria

*A mis padres **Alejandro** y **Nancy** por haberme forjado como la persona que soy en la actualidad, además del esfuerzo y dedicación que han puesto para lograr que sus hijos tengan un futuro mejor*.

*A mis abuelitos **Alberto** y **Rudy** quienes me motivaron a alcanzar todas mis metas, mi abuelita **Ignacia** quien me protege desde mi nacimiento y mi gran amigo **Pablo** con quien compartí extraordinarias aventuras desde los seis años. Todos ellos partieron al cielo, pero desde ahí recibo su apoyo incondicional.*

Agradecimiento

A mis tias ***Gladys*** *,* ***Betty*** *,* ***Pilar*** y mi abuelita **Alejandrina** *quienes siempre nos brindan su atención y cuidados a mis hermanos y a mí desde muy pequeños.*

A mi directora la ***Magister Carmela Velasquez*** *por su apoyo y generosidad en brindarme cartas que me permitieron tener un encuentro formal en las comisarias.*

A mi tía la ***capitana Laura*** *por su tiempo dedicado en acompañarme en las conversaciones con las autoridades de cada comisaria.*

A mi asesor el ***Magister Mario Ninaquispe*** *por su guía y paciencia en la asesoría del desarrollo de mi tesis y culminarla con éxito, gran ejemplo de docente.*

Índice de contenidos

I. Introducción 1
1.1. Introducción 1
1.2. Planteamiento del Problema 2
Determinación del problema 2
Formulación del problema 7
1.3. Objetivos 8
Objetivo principal 8
Objetivos específicos 8
1.4. Importancia y Alcance de la Investigación 8
Alcance 8
Justificación 8
1.5. Limitaciones de la Investigación 9
II. Revisión de la Literatura 10
2.1. Bases Teóricas 10
2.1.1. Problema de Ruteo de Vehículos 10
2.1.2. Optimización Combinatoria 11
2.1.3. Heurística 12
2.1.4. Metaheurística 14
2.2. Antecedentes del estudio 20
Antecedentes Internacionales 20
Antecedentes Nacionales 22
2.3. Marco Conceptual 24
2.3.1. Delito contra el patrimonio 24
2.3.2. Patrullaje Vehicular 24
2.3.3. Herramientas digitales 25
2.4. Definición de términos 25
III. Hipótesis y Variables 26
3.1. Hipótesis 26
Hipótesis General 26
Hipótesis Específicas 26
3.2. Variables 27
3.3. Operacionalización de variables 28
IV. Materiales y Métodos 29

4.1. Área de estudio 29
4.2. Tipo y diseño de investigación 31
4.2.1. Tipo de investigación 31
4.2.2. Diseño de investigación 32
4.3. Población y Muestra 32
Población 32
Muestra no probabilística 32
4.4. Procedimiento, Técnica e Instrumento de recolección 33
4.4.1. Técnica de recolección de datos y información 33
4.4.2. Instrumento de recolección de datos e información 33
4.4.3. Procedimientos de recolección de datos e información 33
4.5. Análisis estadísticos 33
V. Resultados 34
5.1. Método de diseño de rutas empíricas 34
5.2. Adaptación de algoritmos matemáticos para diseñar rutas de patrullaje 41
Diseño de rutas con el algoritmo de Clarke y Wright 43
Diseño de rutas con el algoritmo del Vecino más Cercano 51
Diseño de rutas con el algoritmo de la Búsqueda Tabú 57
Diseño de rutas con el algoritmo Genético 63
5.3. Aplicativo desktop especializado en diseñar rutas de patrullaje vehicular 71
5.4. Análisis de resultados 83
5.4.1. Análisis de resultados para CPNP Piedra Liza 84
5.4.2. Análisis de resultados para CPNP Rímac 96
5.4.3. Análisis de resultados para CPNP Manzano 119
5.4.4. Análisis de resultados para CPNP Ciudad y Campo 142
5.4.5. Análisis de resultados para CPNP Flor de Amancaes 161
5.5. Validación de hipótesis 173
5.5.1. Validacion de hipótesis general 174
5.5.2. Validación de hipótesis especifica 1 175
5.5.3. Validación de hipótesis especifica 2 177
5.5.4. Validación de hipótesis especifica 3 178
5.5.5. Validación de hipótesis especifica 4 180
5.5.6. Validación de hipótesis especifica 5 181
5.6. Propuesta de rutas para patrullaje vehicular 183

VI. Discusión 200
VII. Conclusiones y Recomendaciones 201
7.1. Conclusiones 201
7.2. Recomendaciones 202
Referencias Bibliográficas 203
Anexos 209
Anexo 1. Matriz de Consistencia 209
Anexo 2. Mapa de la jurisdicción de CPNP Piedra Liza 211
Anexo 3. Mapa de la jurisdicción de CPNP Rimac 212
Anexo 4. Mapa de la jurisdicción de CPNP Manzano 213
Anexo 5. Mapa de la jurisdicción de CPNP Ciudad y Campo 214
Anexo 6. Mapa de la jurisdicción de CPNP Flor de Amancaes 215
Anexo 7. Ficha de hoja de ruta 216
Anexo 8. Instrumentos de recolección de datos 217
Anexo 8.1. Puntos críticos del área de jurisdicción 217
Anexo 8.2. Puntos de Protección y Puestos Tácticos 217
Anexo 9. Información Recopilada 218
Anexo 9.1. Comisaria PNP Piedra Liza 218
Anexo 9.2. Comisaria PNP Rímac 219
Anexo 9.3. Comisaria PNP Manzano 220
Anexo 9.4. Comisaria PNP Ciudad y Campo 221
Anexo 9.5. Comisaria PNP Flor de Amancaes 222
Anexo 10. Códigos de programación del aplicativo desktop 223
Anexo 10.1. Formulario Pantalla_inicio 223
Anexo 10.2. Formulario CoordenadasGeograficas 231
Anexo 10.3. Formulario Comisaria 232
Anexo 10.4. Formulario AlgoritmoClarkeYWright 267
Anexo 10.5. Formulario AlgoritmoVecinoCercano 272
Anexo 10.6. Formulario AlgoritmoBusquedaTabu 276
Anexo 10.7. Formulario AlgoritmoGenetico 291
Anexo 10.8. Formulario SolucionesAG 304
Anexo 10.9. Formulario ResultadosFinales 304
Anexo 10.10. Formulario MostrarRutas 308

I. Introducción

1.1. Introducción

La delincuencia es uno de los problemas sociales más recurrentes a nivel mundial, debido a las graves consecuencias que genera a sus victimas. Por tal motivo, los gobiernos del mundo elaboran estrategias de prevención de delitos.

El Perú no es ajeno a que se cometan cotidianamente actividades delictivas, pues las estadísticas indican que el mayor reporte de delincuencia son los delitos contra el patrimonio.

Los delitos más recurrentes en Lima son robos y/o hurtos (Ministerio de Justicia y Derechos Humanos, 2021) , esto motivó al Ministerio del Interior del Perú a publicar un manual de demarcación de territorio jurisdiccional dirigido a todas las comisarias del país, para que en cada área de jurisdicción se haga una división de sectores o subsectores, para asignar vehículos policiales a realizar recorridos constantes sobre el territorio, con la intención de mejorar la actividad de patrullaje.

El distrito del Rímac es uno de los distritos con la mayor cantidad de robos y/o hurtos acontecidos (GESTIÓN, 2019) , siendo el patrullaje vehicular la medida mas recurrente para evitar la comisión de delitos, sin embargo, se toman decisiones empíricas, resultando que los vehiculos realicen largos recorridos. Debido a ello, en la presente investigación se elaboró un aplicativo desktop que brindo propuestas de rutas basado en las soluciones de técnicas matemáticas de la Investigación Operativa.

Esta investigación está organizada de la siguiente manera. El primer capítulo, consta del desarrollo del planteamiento del problema. En el segundo capítulo, se revisó investigaciones previas sobre la problemática planteada y los conocimientos teóricos. En el tercer capítulo, se presentó las hipótesis y variables de la investigación. En el cuarto capítulo, se explica el tipo y diseño de la investigación, ademas de la herramientas usadas en el recojo de información y datos. En el quinto capítulo, se detalla el análisis de los resultados obtenidos. En el sexto capítulo, se realiza la discusión de resultados y comparación con trabajos de investigación de algunos autores y en el último capítulo se exponen las conclusiones y recomendaciones.

1.2. Planteamiento del Problema

Determinación del problema

La delincuencia, es un problema mundial debido a los altos índices de inseguridad que se reportan en cada país, muchas son las causas de este problema, por ejemplo: problemas familiares, problemas económicos, desigualdad social, escasez de recursos, etc. Sin embargo, la gran preocupación son las consecuencias de estos actos, que van desde lesiones leves, secuelas físicas, hasta lesiones mortales que llegan a homicidios o daños psicológicos en sus víctimas. Por esta razón los gobiernos a nivel mundial crearon la seguridad ciudadana.

"La seguridad ciudadana es el proceso de establecer, fortalecer y proteger el orden civil democrático, eliminando las amenazas de violencia en la población y permitiendo una coexistencia segura y pacífica." (Programa de las Naciones Unidas para el Desarrollo, [PNUD], 2014). Dentro de estas amenazas se encuentra la delincuencia, por lo tanto, es obligación de cada estado del mundo devolver la tranquilidad a sus ciudadanos.

Frente a ello, la seguridad ciudadana no solo consiste en eliminar la delincuencia, también en elaborar estrategias de prevención. Durante muchos años se han elaborado planes operativos e incluso modelos de policía y seguridad. Uno de los varios modelos de prevención es el modelo del policial comunitario, creado en los años 70 en Estados Unidos. El modelo se define como "Una manera de pensar, una manera de sentir y obrar que deben tener los policías más próximos a la comunidad y que está basado en un buen trato al público, mayor eficiencia y mejores servicios policiales" (Yépez, 2004, p. 293). El modelo ha tenido varias modificaciones y resultados a lo largo del tiempo en su implementación en cada estado, pero conservando sus características principales. Una de las características del modelo mencionado es la reorientación del patrullaje, la policía no solo va a atender una llamada de emergencia, sino marcar presencia policial en las calles (Bayley David y Skolnick Jerome , 1988, como se citó en Labra Díaz, 2011).

El patrullaje consiste en recorrer zonas geográficas, generalmente para evitar que se cometan delitos. Esta actividad se realiza en un horario y sector definido, por lo que el Estado brinda recursos a las Instituciones involucradas para tal fin. Existen diferentes tipos de patrullaje según su desplazamiento: vehicular, a pie, motorizado etc. Sin

embargo, esta actividad no siempre obtiene los resultados esperados. Un ejemplo, se observó en el experimento de “Patrullaje al Azar de Kansas City” en 1973, donde la ciudad de Kansas fue dividida en tres áreas o sectores. En una de ellas se mantuvo el patrullaje preventivo vehicular, en otra no hubo y en la parte restante el patrullaje se triplicó. Los resultados al cabo de dieciocho meses, indicaron que no había diferencias significativas de la cantidad de delitos ocurridos en las tres partes de la ciudad, con esta experiencia los gobiernos emplean recurrentemente nuevas tácticas para combatir la delincuencia.

Según Labra (2011, p. 57) , la Institución Policial Carabineros de Chile, en la década de los 90 implementó el “Plan Cuadrante de Seguridad Preventiva”, los objetivos fueron dos: potenciar la vigilancia policial preventiva y dinamizar el actuar policial, para dar respuesta oportuna y eficaz a los requerimientos de la comunidad, también se puede señalar que dicho plan está dentro de los principios del modelo de policía comunitario.

El Plan Cuadrante consiste en dividir el área jurisdiccional de una unidad policial (comisaria) en sectores de vigilancia más pequeños denominados cuadrantes, cada cuadrante es custodiado por un vehículo policial u oficial. Esta estrategia resulta hasta ahora ser la técnica adecuada para la prevención de actividades delictivas, sin embargo, se debe de tener en cuenta las dificultades que aparecen en el desarrollo del plan:

- El Plan Cuadrante solamente está establecido en algunos distritos de Chile.
- Limitación de personal en las unidades policiales y recursos logísticos.
- Requiere la elaboración de una adecuada planificación de las operaciones policiales.

Según el portal web Numbeo (base de datos mundial de tasas de delincuencia percibidas), el Perú se ubicó dentro de los 20 países a nivel mundial con mayor índice de criminalidad en el año 2018. El índice de criminalidad es una estimación del nivel de delitos cometidos en una ciudad o país que el Estado tiene conocimiento de que hayan ocurrido. El índice está compuesto de la cantidad de delitos denunciados en homicidios, violencia, robo a personas, agresión en daños físicos, robo de casas, hurtos y robo de vehículos (Quintero Cuello et al., 2008).

Tabla 1

Ranking de países con mayor índice de delincuencia en 2018

Puesto	País	Índice de Criminalidad
1	Venezuela	82.59
2	Papúa Nueva Guinea	79.95
3	Honduras	78.89
4	Sudáfrica	75.71
5	Trinidad y Tobago	72.22
6	Brasil	70.55
7	El Salvador	69.72
8	Bangladesh	68.52
9	Namibia	68.22
10	Siria	67.74
11	Jamaica	66.87
12	Puerto Rico	64.94
13	Perú	63.91

Nota. Se muestra en orden decreciente el índice de criminalidad de cada país según la base de datos de la página web de NUMBEO, los índices menores a 20 son considerados como muy bajos, entre 20 y 40 son bajos, entre 40 y 60 moderados, entre 60 y 80 altos y mayores a 80 muy altos (Numbeo, 2021). Tomado y adaptado de Índice de delincuencia por país 2018, por NUMBEO,2021, https://es.numbeo.com/criminalidad/clasificaciones-por-pa%C3%ADs?title=2018.

Según la Tabla 1 ,el Perú tuvo un índice de criminalidad muy alto, esto se confirma con la publicación de un informe, sobre los índices de criminalidad en el país entre los años 2011 al 2018 del Instituto Nacional de Estadística e Informática (INEI). El documento indica que el 59% de las denuncias en el 2018 son por delitos de hurto y/o robo (GESTIÓN, 2019a).

Debido al contexto en que se encontraba el Perú era necesario implementar nuevas estrategias para combatir la delincuencia a nivel nacional.

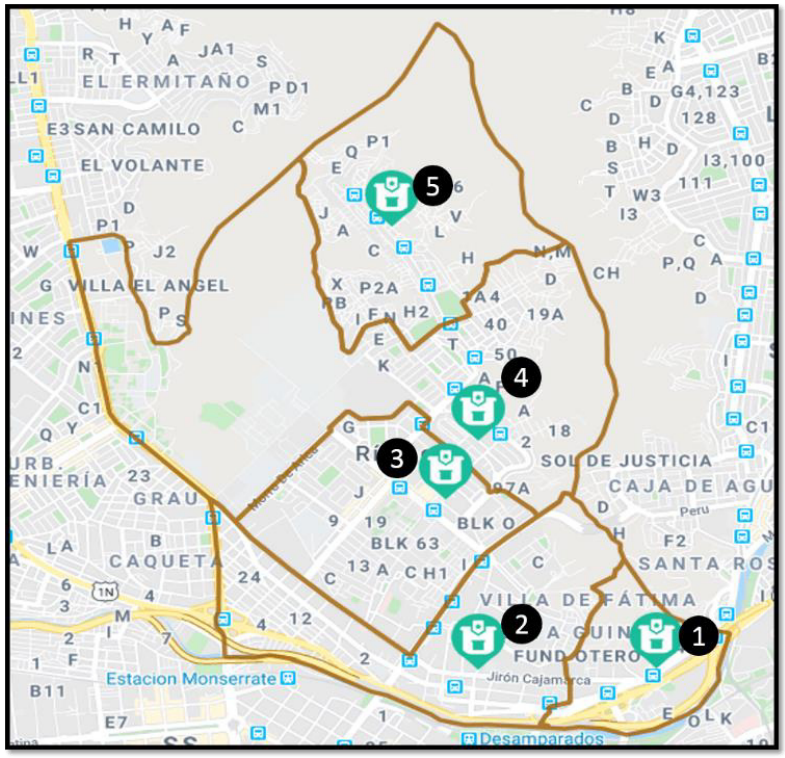
La Dirección General de Seguridad Ciudadana y la Oficina General de Planificación y Presupuesto ambos del Ministerio Interior (MININTER) en conjunto con la sub Comandancia General PNP publicaron una guía llamada: “GUIA METODOLOGICA PARA EL DISEÑO DE SECTORES Y MAPA DEL DELITO EN LA JURISDICCION DE LAS COMISARIAS DE LA POLICIAL NACIONAL DEL PERU” en diciembre del 2018.

En la guía se muestra los lineamientos y criterios técnicos para el diseño de sectores y mapa de delitos, con la finalidad de orientar a las unidades policiales en el patrullaje por sector. Además, se establece como norma que cada comisaria debe de desagregar su jurisdicción en sectores o subsectores y esta información se debe agregar en el Sistema de Denuncia Policial (SIDPL) y el Sistema Informático de Planificación y Control Patrullaje Policial (SIPCOP) (MININTER & Sub Comandancia General PNP, 2018).

La municipalidad del distrito de Rímac, publico en el año 2019: “PLAN ESTRATEGICO DE SEGURIDAD CIUDADANA DEL DISTRITO DEL RIMAC”, este documento muestra que el distrito cuenta con 5 comisarías y se ve el área de jurisdicción de cada una de ellas.

Figura 1

Mapa de área de jurisdicción de las comisarias del distrito de Rímac

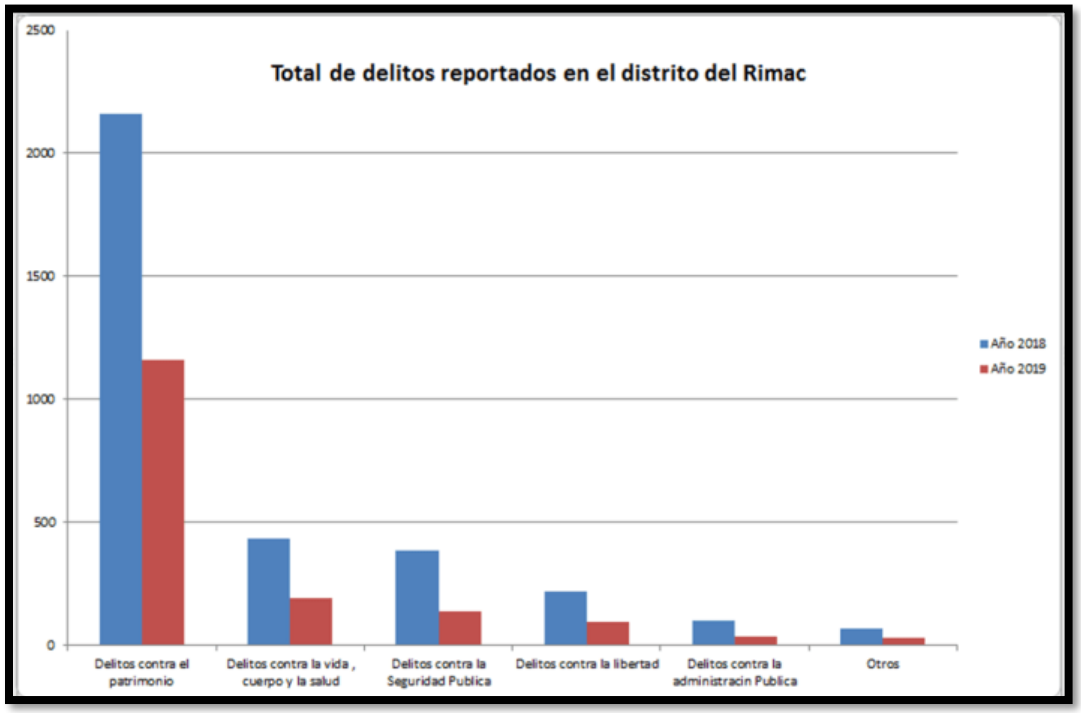


Nota. En la ilustración se muestra la división del distrito, así como el área geográfica correspondiente a cada comisaria. Los puntos oscuros indican la ubicación de una comisaria. Tomado de https://www.mininter.gob.pe/ubica-tu-comisaria.

De la Figura 1, en el punto oscuro (1) se ubica la Comisaria PNP Piedra Liza, en el punto (2) se ubica la Comisaria PNP Rímac, en el punto (3) se ubica la Comisaria PNP Manzano, en el punto (4) se ubica la Comisaria PNP Ciudad y Campo, en el punto (5) se ubica la Comisaria PNP Flor de Amancaes. Además, considerando la guía de división publicado por MININTER, las comisarias dividen su área de jurisdicción en sectores, y asignan a sus vehículos a realizar el patrullaje vehicular.

Figura 2

Gráfico de barras comparativo de delitos reportados en el distrito de Rímac



Nota. La figura muestra un gráfico comparativo de la cantidad de delitos según el tipo de modalidad que fueron reportados en el distrito del Rímac en los años 2018 y 2019. Tomado y adaptado de https://datacrim.inei.gob.pe/ciudadano/

Los resultados de la Figura 2 indican que ha habido una disminución considerable en cada tipo de delito con respecto al año anterior, sin embargo, la cantidad de reportes de delito contra el patrimonio como son el hurto, robo, robo de vehículos, etc., sigue siendo la más alta.

Si bien es cierto, el patrullaje por sectores ha logrado reducir la cantidad de delitos contra el patrimonio, que fue la razón principal para implementar esta estrategia de prevención, no se debe de dejar de lado que existe una gran cantidad de incidentes que

no son reportados en las comisarías, muchos de ellos son hechos recurrentes en puntos específicos del distrito o alrededores de locales vulnerables (colegios, mercados, hospitales, parques). Así mismo durante el patrullaje, los trayectos empleados por los vehículos realizan recorridos extensos porque la elección de una ruta es de manera empírica y muchas veces las trayectorias son las mismas. La decisión de movilizarse empíricamente se debe a la dificultad de planificar rutas de patrullaje o diseñar un conjunto de trayectorias definidas, por lo que trae como consecuencia que el patrullaje no garanticen un recorrido eficiente sobre los puntos críticos que tiene el distrito. En base a lo mencionado anteriormente, se elaboró un aplicativo desktop que desarrolle propuestas de rutas con recorridos cercanos al óptimo sobre los puntos críticos del distrito mediante herramientas de la Investigación de Operaciones.

Formulación del problema

Forma declarativa

La falta de trayectorias definidas en los patrullajes vehiculares en la jurisdicción de cada comisaría del Rímac dificulta el recorrido eficiente por parte de los vehículos policiales, lo cual impide que se pueda optimizar el recorrido de puntos críticos.

Forma interrogativa

¿Cómo optimizar el recorrido de los vehículos policiales en los puntos críticos de las jurisdicciones de las comisarías del distrito del Rímac con metaheurísticas?

Preguntas específicas:

1. ¿Cómo optimizar el recorrido de los vehículos policiales en los puntos críticos de la jurisdicción de la Comisaría PNP Piedra Liza con metaheurísticas?
2. ¿Cómo optimizar el recorrido de los vehículos policiales en los puntos críticos de la jurisdicción de la Comisaría PNP Rímac con metaheurísticas?
3. ¿Cómo optimizar el recorrido de los vehículos policiales en los puntos críticos de la jurisdicción de la Comisaría PNP Manzano con metaheurísticas?
4. ¿Cómo optimizar el recorrido de los vehículos policiales en los puntos críticos de la jurisdicción de la Comisaría PNP Ciudad y Campo con metaheurísticas?
5. ¿Cómo optimizar el recorrido de los vehículos policiales en los puntos críticos de la jurisdicción de la Comisaría PNP Flor de Amancaes con metaheurísticas?

1.3. Objetivos

Objetivo principal

Diseñar rutas para el recorrido óptimo de puntos críticos en las jurisdicciones de las comisarías del distrito de Rímac aplicando metaheurísticas.

Objetivos específicos

1. Diseñar rutas para el recorrido óptimo de puntos críticos en la jurisdicción de la Comisaria PNP Piedra Liza aplicando metaheurísticas.
2. Diseñar rutas para el recorrido óptimo de puntos críticos en la jurisdicción de la Comisaría PNP Rímac aplicando metaheurísticas.
3. Diseñar rutas para el recorrido óptimo de puntos críticos en la jurisdicción de la Comisaría PNP Manzano aplicando metaheurísticas.
4. Diseñar rutas para el recorrido óptimo de puntos críticos en la jurisdicción de la Comisaría PNP Ciudad y Campo aplicando metaheurísticas.
5. Diseñar rutas para el recorrido óptimo de puntos críticos en la jurisdicción de la Comisaría PNP Flor de Amancaes aplicando metaheurísticas.

1.4. Importancia y Alcance de la Investigación

Alcance

En esta investigación se elaboró un aplicativo de computadoras hecho en Visual Studio 2019, para que genere propuestas de rutas de patrullaje vehicular, con el fin de usarlas en la prevención de delitos en el distrito del Rímac. El aplicativo brinda alternativas de rutas para las cinco comisarías del distrito, dichas rutas garantizan que haya recorridos cercanos al óptimo por los puntos críticos puesto que están diseñadas con cuatro algoritmos matemáticos de Investigación de Operaciones, para ello se usó la información de la actividad de patrullaje de enero hasta junio del 2021. El aplicativo solo considera la información que ha sido recopilada en el periodo de tiempo mencionado.

Justificación

Justificación Práctica

El presente trabajo de investigación es de competencia de la Investigación de Operaciones porque la situación problemática previamente explicada, se ajusta al problema del agente viajero, por ello se hizo un aplicativo desktop que utiliza las técnicas de la Investigación de Operaciones para facilitar la construcción de rutas. Por lo tanto,

las comisarias del distrito del Rímac contarán con un aplicativo que muestre propuestas de rutas y estas cumplirán con el recorrido sobre puntos críticos existentes. Todo ello para prevenir actos delictivos.

Justificación Social

La presente investigación está dirigida a ayudar a combatir el problema social de la delincuencia mediante la generación de propuestas de rutas, las cuales se basan en la aplicación de técnicas de la Investigación de Operaciones, estas rutas ayudarán a que los vehículos realicen recorridos cercanos al óptimo por los puntos críticos del distrito de Rímac con el fin de prevenir actos delictivos.

Justificación Teórica

La utilización de técnicas de la Investigación de Operaciones, además de los conceptos básicos del problema de ruteo de vehículos, patrullaje policial y puntos críticos, son base para diseñar las rutas de patrullaje. Para la elaboración de rutas se utilizan heurísticas y metaheurísticas, considerando como destinos los puntos críticos de la zona a custodiar. Los resultados obtenidos servirán para ver si es factible desarrollar rutas policiales con las técnicas matemáticas mencionadas y puntos críticos, sin embargo, también será discusión si es adecuado considerar otros factores o escenarios para la solución del problema a abordar en esta investigación. Ello le servirá como apoyo al investigador para trabajos futuros relacionados al problema de ruteo de patrullas policiales.

1.5. Limitaciones de la Investigación

La información requerida para el presente trabajo de investigación es confidencial, por ello se realizó las gestiones con las autoridades pertinentes, para obtener el permiso de utilizar los registros de patrullaje con fines académicos.

Con respecto al recorrido diario que realizan los patrulleros, este se encuentra anotado en un “Cuaderno de Rutas” de manera genérica. Sin embargo, no se precisa la trayectoria que hayan realizado los vehículos y el criterio usado para la elección de los puntos críticos a recorrer, debido a que se toman decisiones de manera empírica. Además, el recorrido no es registrado en una base de datos, por lo que para fines de esta investigación se planteó un método para replicar rutas empíricas.

II. Revisión de la Literatura

2.1. Bases Teóricas

2.1.1. Problema de Ruteo de Vehículos

El problema de ruteo de vehículos es conocido en la literatura científica como Vehicle Routing Problem (VRP). Es uno de los problemas más estudiados debido a la importancia de su aplicación en las actividades logísticas.

La descripción clásica del VRP consiste en que varios vehículos comienzan su recorrido en un centro de distribución y deben visitar a clientes para realizar entregas, al finalizar los camiones deben de retornar al centro de distribución. Dependiendo de características como: número de vehículos disponibles, capacidad de los camiones, incorporación de ventanas de tiempo, entre otras restricciones, estas generan diferentes variantes del problema. La solución del problema consiste en determinar una ruta o un conjunto de rutas que satisfagan a las restricciones mencionadas,las mejores rutas serán aquellas que cumplan una función objetivo. Algunas de las funciones objetivo son: minimizar costos fijos, minimizar el número de vehículos, minimizar tiempo total de transporte y/o distancia total recorrida entre otras funciones (Garcés & Zapata, 2019).

2.1.1.1. Problema del Agente Viajero

Conocido en la literatura científica como Traveling Salesman Problem (TSP), el problema se describe “dado un numero n de ciudades, las cuales poseen características de costos o distancias, el vendedor partiendo de una ciudad debe recorrer todas las demás exactamente una vez hasta retornar a su punto de partida”(Lopez et al., 2008, p. 399). En este tipo de problema no es considerada la demanda de los clientes y restricciones temporales. De esta manera el “TSP tiene que ver con hallar el recorrido mas corto (cerrado) en una situación de *n* ciudades, donde cada ciudad es visitada una sola vez antes de regresar al punto de partida” (Taha, 2012, p. 395).

El TSP es una generalización de varios de los tipos del problema de ruteo de vehículos, es por ello que se le considera el más sencillo de plantear, sin embargo, este pertenece a la clase de problemas NP-Hard y es considerado dentro de los problemas de optimización combinatoria (Benavidez, 2012 , como se citó en Garcés & Méndez, 2019, p. 27).

2.1.1.2. Problema de Ruteo de Vehículo con Capacidad

Conocido en la literatura científica como Capacitated Vehicle Routing Problem (CVRP), en este tipo de problema se toma en cuenta la restricción de capacidad de carga de mercancías del vehículo para buscar la solución.

La descripción del CVRP consiste en que una flota de vehículos con capacidad homogénea tiene que atender las demandas determinísticas de los clientes. La solución del problema es determinar las rutas para los vehículos con la mínima distancia total de recorrido, la única restricción es que la suma de la demanda de los clientes pertenecientes a una ruta no debe de exceder a la capacidad del vehículo (Sánchez, 2015, p. 13).

2.1.2. Optimización Combinatoria

La optimización es el mejoramiento de una actividad para que el rendimiento sea más eficiente. Los problemas de optimización tratan de elegir una decisión óptima para maximizar ganancias, velocidad, eficiencia, etc., o minimizar costos, tiempo, riesgo, error, etc. Se añade también que “el proceso de optimización en la resolución de problemas consiste en mejorar alguna solución relativamente buena conseguida previamente o incluso llegar a conseguir una solución óptima” (Vargas & Penit , 2016, p. 6).

Una de las ramas de la optimización es la optimización combinatoria, esta afronta aquellos problemas con un número finito de soluciones. “En estos problemas el objetivo es encontrar el máximo (o mínimo) de una determinada función sobre un conjunto finito de soluciones que denotaremos por S” (Martí, 2003, p. 2). El conjunto S generalmente es muy extenso de explorar en la búsqueda de una buena solución, por lo que la aplicación de algoritmos matemáticos facilita poder encontrar una solución en un tiempo prudente. Según Arteaga (2013) la optimización combinatoria elabora algoritmos que puedan solucionar problemas de optimización que tengan la particularidad de tener un número extenso de soluciones factibles.

“Un algoritmo es una serie de pasos organizados que describe el proceso que se debe seguir, para dar solución a un problema específico.” (Fadul, 2004, p. 14). El algoritmo debe indicar con precisión la realización de cada paso, además al ejecutarse varias veces se debe obtener el mismo resultado en cada intento siempre y cuando sean

las mismas entradas y se debe terminar en algún momento porque hay un numero finitos pasos.

Los algoritmos de optimización combinatoria logran reducir el tamaño de búsqueda del espacio de soluciones, de modo que la exploración sea más eficiente (Hoyos & Rubio, 2012, p. 32). Algunos de estos algoritmos son las heurísticas y metaheurísticas.

En este trabajo de investigación se desarrolla la teoría y procedimientos correspondiente a las heurísticas y metaheurísticas que resuelven el CVRP y TSP.

2.1.3. Heurística

Las Heurísticas son técnicas que realizan procedimientos sencillos para encontrar rápidamente buenas soluciones a problemas de optimización combinatoria (Taha, 2012).

Los procedimientos de las heurísticas reducen el espacio de búsqueda de soluciones eliminando caminos que lleven a soluciones de peor calidad, sin embargo, presentan el inconveniente de que algunos de los caminos eliminados puedan llevar a una solución global, por lo tanto las soluciones encontradas serían soluciones locales que no son la mejor solución posible (Rabanal, 2010). Sin embargo una heurística bien estructurada puede brindar soluciones cercanas al óptimo (Hillier & Lieberman, 2010).

2.1.3.1. Algoritmo del Vecino más cercano

Es un algoritmo heurístico con el método más sencillo para encontrar solución al CVRP y TSP. Este algoritmo está basado en ir construyendo rutas según la cercanía de puntos. Antes de la ejecución del algoritmo se considera que existen clientes y un depósito que se encuentran dispersos dentro de un área geográfica.

El procedimiento de este algoritmo para resolver el CVRP según Contreras y Díaz (2010) es:

1. Ubicarse en el punto de partida o depósito para construir una nueva ruta.
2. Tomar la distancia desde el punto agregado recientemente en la ruta actual hacia cada uno de los demás puntos restantes que no estén incluidos en alguna ruta.
3. Se elige el punto que tenga la menor distancia entre él y el último punto agregado en la ruta actual. Si existe empate, se elige cualquiera arbitrariamente. El punto elegido se añade a la ruta actual.

4. Repetir el paso 2 hasta que cumpla las restricciones asociadas al CVRP o si se satisfacen la restricción de capacidad.
5. Después de haber construido una ruta, unir el último punto añadido de la ruta actual con el punto de partida, para indicar que al final de la trayectoria se debe regresar al punto donde partió.
6. El algoritmo termina si todos los puntos están incluidos en alguna ruta.

2.1.3.2. Algoritmo de Clarke y Wright

El algoritmo de Clarke y Wright o método del ahorro es una heurística popular para resolver el CVRP. Se identifica al depósito o punto de partida (0) con los clientes o destinos (i) o (j) y estos son denotados con los índices de $(1,2,\dots,n)$, los costos de trasladarse de un destino (i) al destino (j) son conocidos. El procedimiento es la de combinar dos rutas distintas $(0,\dots,i,\dots,0)$ y $(0,\dots,j,\dots,0)$,para formar una nueva ruta $(0,\dots,i,j,\dots,0)$ mediante el criterio del ahorro. Según (Clarke & Wright, 1964), los ahorros se calculan con la ecuación 1.

$$S_{ij} = C_{i0} + C_{0j} - C_{ij} \qquad (1)$$

Donde:

- C_{0j} , representa el costo de trasladarse del punto de partida (0) hacia el destino (j).
- C_{i0} , representa el costo de trasladarse del destino (i) hacia el punto de partida (0).
- C_{ij} , representa el costo de trasladarse del destino (i) hacia el destino (j).

Se tiene en cuenta que en la ecuación 1, S_{ij} debe ser mayor o igual a cero, representando la distancia ahorrada por el cambio. Despues de calcular ahorros, según la propuesta de Olivera (2004) el algoritmo puede construir rutas de forma paralela (de forma simultánea) o de forma secuencial (de una por vez). Para el presente tema de estudio se utilizó la propuesta de algoritmo de Clarke y Wright versión secuencial, siendo este el procedimiento:

- Paso 1 (inicio). Para cada cliente (i) construir la ruta $(0,i,0)$
- Paso 2 (cálculo de los ahorros). Calcular S_{ij} para cada par de clientes (i) y (j).

- Paso 3 (Selección). Si todas las rutas fueron consideradas, terminar. Si no, seleccionar una ruta que aún no haya sido considerada.
- Paso 4 (Extensión). Sea $(0, i, \ldots\ldots, j, 0)$ la ruta actual. Si no existe ningún ahorro conteniendo a (i) o a (j) , ir a paso 3. Sea S_{k^*i} $(o\ Sjl^*)$ el máximo ahorro conteniendo a (i) o a (j). Si (k^*) o (l^*) es el último (o primer) cliente de su ruta y la combinación de dicha ruta con la actual es factible, realizar dicha combinación. Eliminar S_{k^*i} o S_{jl^*} de futuras combinaciones. Ir a paso 4.

2.1.4. Metaheurística

Las metaheurísticas son estrategias que diseñan o mejoran los procedimientos de las heurísticas para alcanzar un mejor rendimiento en la búsqueda de soluciones. Las soluciones de las metaheurísticas son buenos resultados, pero no son óptimos sino soluciones sub óptimas (Hoyos & Rubio, 2012). Por este motivo, son algoritmos aproximados a la optimización , los procedimientos son iterativos de forma inteligente, que guían a una heurística hacia una mejor exploración y explotación dentro de un espacio de soluciones (Castañeda, 2013) .

Existen características que han sido descritas en las revisiones literarias referente a metaheurísticas, según Voss (2001) algunas de ellas son:

1.- Durante el proceso de búsqueda, las metaheurísticas pueden incorporar estrategias para evitar la exploración de regiones, donde no sea prometedora encontrar una buena solución.

2.- Para encontrar soluciones de alta calidad, se utiliza la información específica de un problema y/o se aprende de los procedimientos ejecutados a lo largo del proceso.

2.1.4.1. Algoritmo Búsqueda Tabú

Conocido en la literatura científica como Tabu Search (TS), es una metaheurística clásica para resolver problemas de optimización combinatoria. Esta metaheurística utiliza procedimientos con “ideas de sentido común para permitir que el proceso de búsqueda escape de un óptimo local.”(Hillier & Lieberman, 2010, p. 570). Estos procedimientos son de aprendizaje, se usan estrategias especiales para la búsqueda de soluciones.

Existe una extensa literatura referida a procedimientos de este algoritmo para resolver diversos problemas de optimización combinatoria, sin embargo, para esta investigación se usó la teoría correspondiente al método para resolver el TSP. Se

presenta la descripción del procedimiento del TS para la resolución del TSP planteado por Taha (2012).

La ruta es representada como 0-1-2-3-4-5-0, siendo (0) el punto de partida y (1,2,3,4,5) los destinos. Un segmento de la ruta consiste en seleccionar dos destinos adyacentes, para el ejemplo sus segmentos son: (0-1), (1-2), (3-4), (4-5), (5-0). El costo de la ruta es identificado como la solución actual del problema y es denotado por s.

El procedimiento de TS consiste en que la iteración *k* se mueva de la solución de s_k a s_{k+1} , esta última es la mejor solución de un subconjunto de soluciones vecinas de s_k , denotado por $N(s_k)$. No necesariamente s_{k+1} tiene menor costo que s_k. Se utiliza el instrumento llamado *lista tabú* (L_k) para recordar los movimientos de la búsqueda anterior y deshabilitarlos durante un *periodo de tenencia* específica (τ), todo ello para que, durante la búsqueda, exista prohibiciones temporales de volver a examinar soluciones anteriores. Las soluciones prohibidas se llaman *solución tabú*, el segmento que lleva a las soluciones tabú se llaman *movimiento tabú*. Se utiliza el *criterio de aspiración*, para aceptar una solución que, aunque sea solución tabú, mejora el costo de la solución encontrada hasta el momento. En resumen, los elementos que están presentes en el procedimiento del TS son:

k = iteración del algoritmo

s_k = Solución de la iteración k

$N(s_k)$ = Vecindad de s_k o subconjunto de soluciones vecinas de s_k

L_k = Lista tabú en la iteración k

τ = Periodo de tenencia en la lista tabú

Para el TSP, existen los siguientes principios del TS:

Recorrido de inicio: Consiste en determinar una solución inicial que será utilizada como punto de partida para buscar una mejor solución. El autor propone cuatro opciones, pero para esta investigación la solución inicial será la obtenida por el algoritmo del vecino más cercano.

Inversión de un subrecorrido: Consiste en realizar una inversión a un subrecorrido de una ruta (por ejemplo, la ruta 0-1-2-3-4-0 tiene un subrecorrido 1-2-3) e identificar qué segmentos han sido eliminados y qué segmentos han sido añadidos para producir una nueva ruta. Considerando como ejemplo la ruta 0-1-2-3-4-5-0. La inversión del

subrecorrido 3-4-5 produce la ruta 0-1-2-5-4-3-0 cuando de la ruta 0-1-2-3-4-5-0 se elimina los segmentos (2-3) y (5-0) y agregando los segmentos (2-5) y (3-0). El número mínimo de subrecorridos es 2 (por ejemplo 1-2 o 2-3) y el máximo es (*n – 2)* para una matriz de distancia simétrica.

Vecindad en la iteración *k*: Es el subconjunto de rutas que han sido generadas por la aplicación de inversión de subrecorridos en la iteración *k*.

Movimiento tabú: Una ruta de $N(s_k)$ es tabú si sus dos segmentos eliminados están en la lista tabú

Siguiente movimiento en la iteración *k*: Consiste en identificar la ruta con el recorrido más corto de $N(s_k)$. Seleccionarla como la mejor solución de la iteración *k* si no es tabú, o si es tabú, pero produce una mejor solución (criterio de aspiración). En caso contrario, excluir el recorrido corto y repetir el análisis con la siguiente ruta del $N(s_k)$ más corto.

Periodo de tenencia tabú τ en la iteración *k:* Es el número de iteraciones sucesivas que un elemento permanece en la lista tabú.

Cambios en la lista tabú en la iteración *k*: Los segmentos de la ruta que son añadidos para producir s_k del s_{k-1} son agregados a la lista tabú y a los segmentos que estaban en la lista tabú se les quita una unidad en su periodo de tenencia o (τ-1). Los segmentos de la lista tabú que completan el periodo de tenencia son eliminados de la lista.

Los pasos para el algoritmo de la búsqueda tabú son:

- Paso 0. Seleccionar un *recorrido de inicio* s_0 donde k=0 y determinar *el periodo de tenencia en la lista tabú*.
- Paso 1. Realizar la *inversión de subrecorridos*, luego determinar de la *vecindad en la iteración k* aquellos que no sean *movimiento tabú*.
- Paso 2. Realizar *siguiente movimiento en la iteración k* y hacer *cambios en la lista tabú en la iteración k*.
- Paso 3. Si se llega a una condición de terminación (realizar una cantidad determinada de iteraciones), detenerse. Si no realizar una nueva iteración o k = k+1 y volver al paso 1.

2.1.4.2. Algoritmo Genético

El algoritmo genético (AG) es una metaheurística que imita a la evolución biológica. Esto consiste según Hernández (1998), en que los seres vivos durante el proceso de adaptación al medio en el cual buscan desarrollarse tienen en cuenta una medición de dicha adaptabilidad, estas les permite ir transmitiendo a su descendencia características que aportan a que sean más aptas y eliminando aquellas características que obstaculizan su desarrollo. De esta manera los seres vivos más aptos sobreviven y los de peor adaptación se van extinguiendo en generaciones.

Según Hillier y Lieberman (2010) los procedimientos del AG permiten explorar con mayor intensidad la región factible y evolucionar gradualmente, para encontrar las mejores soluciones factibles. Estos procedimientos están basados "en una emulación de la evolución natural de los seres vivos, trabajando sobre una población de soluciones potenciales evoluciona de acuerdo a interacciones y transformaciones únicas" (Nesmachnow, 2004). En esta medida el AG redefine conceptos de la evolución biológica como gen, cromosoma, población entre otros, para incorporarlos en los procedimientos de resolución en los problemas de optimización.

En esta investigación se usaron los conceptos redefinidos de Taha (2012) para el AG, así como el procedimiento para resolver el TSP.

Las soluciones factibles de un problema son llamados "*cromosomas*", este es codificado por un conjunto de "*genes*". Los códigos genéticos suelen ser representados de manera binaria (0,1) o numérica (0,1,2, …, n). El conjunto de *N* soluciones factibles o el conjunto de *N* cromosomas se le conoce como "*población*". La "*aptitud*" de un cromosoma se evalúa mediante la función objetivo de un problema de optimización. El cromosoma con más aptitud brinda un mejor valor de la función objetivo.

El procedimiento general del AG es seleccionar dos "*padres*" de una población, los códigos de genético de cada padre se "*cruzan*" y "*mutan*" (posiblemente) para engendrar dos "*hijos*". Estos hijos reemplazan a los dos cromosomas menos aptos de la población y se vuelve a repetir el procedimiento de selección de dos padres.

Para resolver el TSP se añaden detalles adicionales al proceso general del AG:

Codificación de genes: La ruta es representada por el código genético de tipo numérico. Sea 1-2-3-4-5-1 la representación de una ruta, siendo (1) el punto de partida y (2,3,4,5) los destinos.

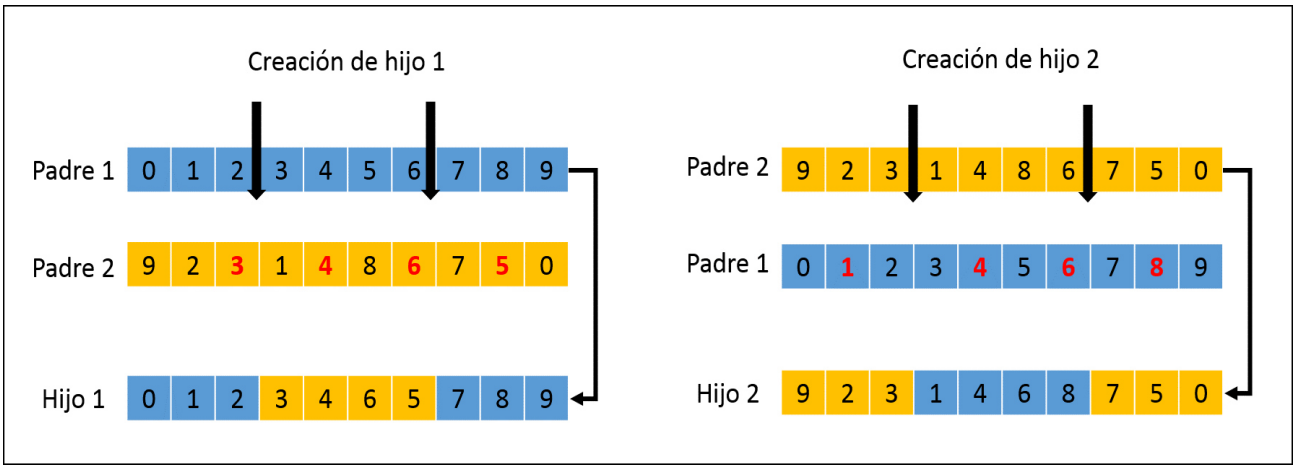
Población inicial: Consiste en generar rutas para el conjunto de la población inicial. Una ruta se construye especificando un punto de origen, luego se selecciona un destino (no debe estar incluido en el recorrido actual) que pueda llegarse a él a través de un segmento de recorrido finito partiendo desde el último punto que fue incluido en la trayectoria actual. El punto seleccionado es añadido a la derecha del último destino de la ruta actual y el mismo proceso se repite hasta que todos los destinos estén incorporados a la ruta. Para la presente investigación, se considera que todos los destinos son adyacentes entre sí, por lo que el proceso de construcción de una ruta (o un cromosoma) de una población inicial siempre va a generar uno con recorrido factible.

Creación de un hijo: Consiste en seleccionar dos padres P1 y P2 de la población, para cruzar sus genes y crear dos hijos C1 y C2. La selección de P1 y P2 consiste en identificar las rutas con la mínima longitud de recorrido del conjunto de población. Existen varias técnicas para el cruzamiento de genes, el mismo autor utiliza la técnica el “Cruce Ordenado”, sin embargo, para esta investigación se usó el procedimiento del “Operador de Cruce Ordenado con Varios Puntos”.

Para interés del presente trabajo se va explicar el desarrollo del operador de cruce ordenado con dos puntos. Según Sanz (2016), se elige aleatoriamente los puntos de corte del cromosoma de los padres por donde se va a efectuar el cruce e intercambio de genes. Del primer padre (P1), se seleccionan los genes consecutivos que están contenidos entre el inicio y final de los cortes, estos son identificados en el padre 2 (P2). Los genes del P1 restantes descienden al hijo 1 (C1) en el mismo orden y los genes del padre 2 (P2) que fueron identificados son insertados en C1 en el mismo orden en que aparecen. El segundo hijo (C2) es formado repitiendo el mismo procedimiento. Del segundo padre (P2), se seleccionan los genes consecutivos que están contenidos entre el inicio y final de los cortes, estos son identificados en el padre 1 (P1). Los genes del P2 restantes descienden al hijo 2 (C2) en el mismo orden y los genes del padre 1 (P1) que fueron identificados son insertados en C2 en el mismo orden en que aparecen. En la Figura 3 se muestra lo explicado.

Figura 3

Cruzamiento del operador de cruce por dos puntos

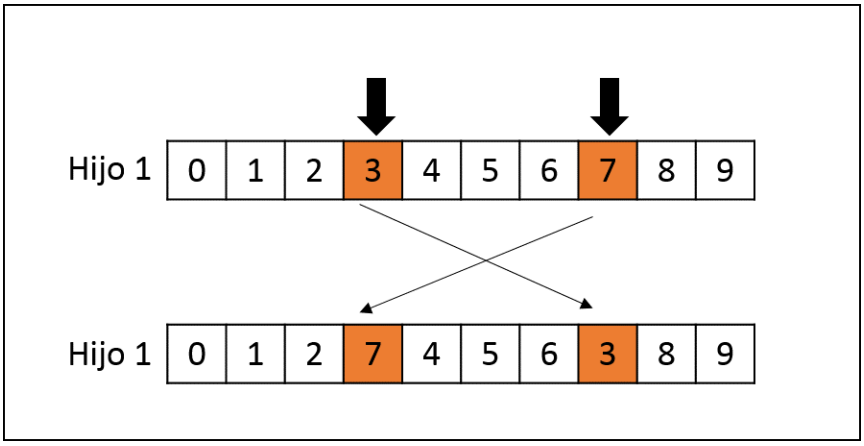


Nota. Se muestra un ejemplo de cruzamiento de los genes de los padres.

Mutación: Cada hijo tiene una probabilidad de mutación denotada por *R*. Por ello si R < 0.1, entonces existe una mutación en los genes del hijo. La mutación será al intercambiarse dos genes, mediante la selección al azar de dos posiciones del código genético del cromosoma (es excluido el punto inicial) de la ruta. En la Figura 4 se muestra lo explicado.

Figura 4

Mutación de hijo por intercambio de posición



Nota. La figura muestra un ejemplo de mutación del cromosoma de un hijo.

Los pasos para el algoritmo genético son:

Paso 0:

- Generar una población aleatoria con *N* cromosomas factibles. Los cromosomas serán las rutas factibles iniciales.
- Para cada cromosoma de la población, evaluar su aptitud. La aptitud para el TSP es el valor de la distancia de cada ruta. Registrar el cromosoma con la mejor solución, denotado por s^* .

Paso 1:

- Seleccionar dos cromosomas padres de la población. Los dos padres deben ser las rutas con la mínima longitud de recorrido del conjunto de población.
- Cruzar los genes de los padres para crear dos hijos.
- Mutar (si es posible) los genes de los hijos.
- Reemplazar los cromosomas más débiles de la población por los nuevos hijos y actualizar el valor de s^*.

Paso 2:

- Si existe una condición de parada, detenerse y s^* es la mejor solución. Si no, repetir el paso 1.

2.2. Antecedentes del estudio

Antecedentes Internacionales

La investigación de H. Chen, T. Cheng y S. Wise (2015), tuvo como objetivo la elaboración de un modelo que construya rutas diarias a vehículos policiales para patrullar segmentos de calles con alto índice delictivo. Esta situación fue adaptada como un "Problema del cartero rural", lo cual consistió en encontrar la secuencia de ruta más corta hacia un subconjunto de segmentos pertenecientes a un grafo no dirigido conexo. La construcción de las rutas está basada en la estrategia bayesiana y el algoritmo de la colonia de hormigas probabilística, además se tuvo en cuenta que el vehículo debe evitar patrullar el mismo punto que sus compañeros de equipo. El modelo fue simulado en el programa MASON, utilizando datos empíricos y datos de incidentes de delitos reales teniendo como área de estudio Camden Borough en Inner London. Los resultados indicaron que la estrategia fue eficaz y permitió que la ruta no sea predecible para un individuo. La investigación tuvo como conclusiones incluir en el modelo respuestas a situaciones de emergencia que pueda atender un patrullero y personalizar el patrullaje para controlar un tipo de delito específico en lugar de delitos generales. La semejanza

del presente trabajo de investigación con lo expuesto anteriormente, es el diseño de rutas a patrullas policiales utilizando datos reales de un área determinada, sin embargo, se va a utilizar una metodología distinta en la construcción de rutas, puesto que se utilizó cuatro técnicas matemáticas para la creación de varias alternativas de rutas que puedan ser usadas en el patrullaje de puntos críticos.

La investigación de Yassen, Arram, Ayob y Nazri (2017) tuvo como objetivo proponer un modelo de solución del problema de ruteo de patrullas policiales (PPRP), basado en el problema de ruteo de vehículos con ventanas de tiempo (VRPTW),se realizaron cambios en los elementos clásicos del VRPTW , los vehículos pasaron a ser las patrulleros policiales, los clientes son los puntos críticos, la ventana de tiempo del cliente es el periodo potencial de delincuencia en el punto crítico, el tiempo de servicio del cliente se convirtió en el tiempo de parada del vehículo en el punto crítico, la capacidad del vehículo fue considerada como el tiempo máximo de patrullaje del vehículo por día y la demanda del cliente no fue tomada en cuenta. El modelo consiste en la utilización de la “Heurística Codiciosa de Mayor Prioridad” y la “Heurística Codiciosa del Vecino más Cercano”. La primera se usó para seleccionar los puntos críticos de mayor prioridad y la segunda para la elaboración de rutas. El modelo fue simulado con los datos que se usó en el trabajo de investigación de Solomon, dicho trabajo consistió en el diseño y análisis de algoritmos para problemas de generación de rutas con restricciones de ventana de tiempo (Solomon, 1987).Los resultados del modelo indicaron que los puntos críticos más cubiertos fueron aquellos con alta prioridad y que la utilización de la heurística codiciosa del vecino más cercano fue más eficiente para construir una solución que la heurística codiciosa de mayor prioridad. La conclusión a la que se llego fue de incentivar la realización de trabajos futuros basados en mejorar la solución obtenida mediante la implementación de un algoritmo de mejora iterativa. La semejanza con el presente trabajo de investigación es el uso de un cambio hecho al elemento del VRPTW para la construcción de rutas a vehículos policiales, siendo los puntos críticos como los clientes a visitar por los vehículos, sin embargo, no se va a considerar lo referido a seleccionar puntos críticos que tenga mayor prioridad además en el diseño de las rutas se usaron dos heurísticas y dos metaheurísticas.

El trabajo de Gamba, Domínguez, Navarini, Díaz y Lombardo (2019), tuvo como objetivo desarrollar una APP. La APP genera rutas de patrullaje, para que se recorra el mayor número de puntos de delitos posibles, debido a ello se busca un equilibrio entre la distancia en kilómetros a recorrer y la cantidad de puntos. La investigación está desarrollada mediante técnicas en Inteligencia Artificial, más precisamente se usó el Algoritmo Genético para la generación de rutas, además la APP fue programado en el lenguaje de programación C#, Python y se utilizó una base de datos MySQL. Se tuvo como referencia los datos proporcionado por la Comisaria 1era de la ciudad de Ramallo, provincia de Buenos Aires (Argentina). Los resultados indicaron que hubo un balance entre la distancia recorrida y la cantidad de puntos delictivos que visita un patrullero. La investigación tuvo la conclusión de expandir su aplicación a otras regiones, así mismo agregar otras funcionalidades al APP como la actualización de rutas en tiempo real. La similitud de la investigación en curso con el trabajo expuesto es el desarrollo de rutas de patrullaje dentro de un aplicativo desktop mediante la aplicación del algoritmo genético, sin embargo, en la investigación en curso se uso cuatro técnicas matemáticas para asegurar diversas alternativas de rutas de patrullaje.

Antecedentes Nacionales

La tesis de Flores (2017) describió el problema de la deficiencia del patrullaje integrado entre la policía nacional del Perú y el serenazgo en la jurisdicción de la comisaria PNP Tarapoto, debido a la baja frecuencia de patrullaje en las zonas de riesgo. El objetivo del trabajo fue analizar el impacto de aplicar la inteligencia de negocios en el patrullaje integrado. Se hizo un estudio para obtener información de las rutas diarias que realizan los patrulleros y la frecuencia del patrullaje, en las zonas de riesgos durante tres meses. Se encontró que los vehículos policiales realizan recorridos con una frecuencia de 23 veces al día. Luego se utilizó la inteligencia de negocios para obtener datos estadísticos y estos usarlos para mejorar el patrullaje integrado. Los resultados posteriores a la implementación de la inteligencia de negocios indicaron que la frecuencia de patrullaje aumento a 32 veces al día. La tesis concluyo que sería ideal que se actualice el mapa de delitos semanalmente con el fin de ir mejorando los resultados obtenidos. Esta tesis tiene semejanza con la presente investigación en mejorar la actividad de patrullaje vehicular, pero se utilizaron técnicas matemáticas para tal fin.

Otra propuesta en la prevención de delitos fue hecha por Gutierrez (2018) , en su tesis presento la problemática que se tiene para combatir la delincuencia con el patrullaje vehicular, porque los patrulleros no consideran el índice de criminalidad y ello origina una mala distribución de vehículos en las calles, lo que ocasiona la poca prevención de actos delictivos. Debido a ello el trabajo tuvo por objetivo el desarrollo de un sistema aplicativo o página web para que los usuarios ingresen información de delitos informales o no reportados en las comisarías, luego con la información registrada y el algoritmo de GRASP, se presenta una propuesta de distribución cercano al óptimo de patrulleros en los lugares donde se hayan registrado los delitos. Considerando como paso previo la determinación de bloques de longitud de 1 km para cada distrito, dichos bloques sirven para localizar los incidentes registrados y facilitar la asignación de un vehículo policial de alguna comisaria que esté más cerca. Se realizo una prueba con los datos de delitos basados en noticias del distrito de Surquillo, obteniendo como resultado tres soluciones que brindo una asignación óptima. La conclusión de esta tesis es realizar la prueba en otros distritos y conectar la aplicación con la base de datos de la Policía Nacional del Perú. Esta tesis se relaciona con la investigación en curso, porque se busca proponer una alternativa de mejora en las actividades de patrullaje vehicular, sin embargo, en la investigación en curso la actividad es el diseño de rutas los delitos como robo y/o hurto registrados en las comisarías del distrito del Rímac.

El trabajo de investigación de Zevallos (2019) , presenta una propuesta de mejora al sistema de patrullaje en el distrito de San Miguel, considerando actividades como patrullaje vehicular , patrullaje motorizado y patrullaje a pie. Se tomó como referencia el modelo de programación lineal entera “Maximum Covering and Patrol Routing Problem” (MCPRP), este modelo matemático consiste en determinar rutas a los vehículos policiales, para cubrir una gran cantidad de lugares donde ocurren choques de vehículos con mucha recurrencia, debido a que se estima que la presencia policial sobre estos puntos reduce el número de accidentes (Keskin et al., 2012). Con el uso del MCPRP y algunas modificaciones, se buscó maximizar la cantidad de tiempo prestada para el recorrido de puntos críticos que serán patrullados por diferentes vehículos dentro de una ventana de tiempo, los puntos críticos que fueron considerados son los delitos cometidos en la vía pública y para definir la ruta a seguir para los turnos de patrullaje se tomó en

cuenta la división de los sectores definidas por la municipalidad. EL modelo con la información requerida del distrito fue ejecutado con el software “IBM ILOG CPLEX”, el cual brindo los siguientes resultados: La asignación a cada sector del tipo de sereno u oficial y rutas de patrullaje propias. La conclusión es agregar al modelo matemático restricciones que cubran las necesidades de cada sector del distrito, por ejemplo, cubrir robo de transeúntes, robos de vehículos, etc. Este trabajo es pertinente con la investigación aquí planteada, ya que también está enfocado en el diseño de rutas para patrullaje en un distrito de Lima, sin embargo, la presente investigación está enfocado en proponer rutas solamente orientados al patrullaje vehicular con la aplicación de heurísticas y metaheurísticas.

2.3. Marco Conceptual

2.3.1. Delito contra el patrimonio

Un delito es la conducta realizada por una persona que va en contra de las normas y leyes impuestas por la sociedad, de cometerlo será castigado con una pena o sanción según el código penal de cada país. Un tipo de delito son los delitos contra el patrimonio, que son “aquéllos destinados a menoscabar el activo de bienes y derechos de un particular, persona jurídica o institución pública, con ánimo de lucro, ya sea propio o en beneficio de un tercero.” (G.Elías & Muñoz, 2021), es común que estos tipos de delitos atenten contra los bienes de una persona, causando en ella un daño físico o psicológico, este es castigado por una pena privativa de la libertad. Los delitos más comunes son el robo y hurto.

- Hurto: Se le denomina a la acción del apoderamiento ilícito de un objeto ajeno, pero a diferencia del robo se realiza sin ningún acto de violencia (Real Academia Española, 2020a).
- Robo: Se le denomina a la acción del apoderamiento ilícito de un objeto, utilizando la violencia o intimidación a la víctima además del uso de la fuerza para acceder a un bien (Real Academia Española, 2020b).

2.3.2. Patrullaje Vehicular

El concepto general de patrullaje es de recorrer una zona geográfica para evitar que se cometan delitos. La definición según el Ministerio de Defensa Nacional Policía Colombiana (2008), es la actividad de vigilancia urbana para el desarrollo de acciones

preventivas, disuasivas y control. La finalidad es neutralizar la comisión de delitos y contravenciones además de incrementar la percepción de seguridad. En la planeación de la realización del patrullaje, se debe de considerar el sector o el terreno a cubrir con el servicio de vigilancia y la duración del patrullaje, este es realizado por turnos y no debe de exceder de ocho horas continuas. Existen varias clases de patrullaje sea por desplazamiento o horario.

La misma institución colombiana detalla que el patrullaje vehicular está basado en el desplazamiento de un vehículo o motocicleta en zonas urbanas. Se añade también lo descrito por el Cuerpo de Policía Nacional Bolivariana de Venezuela (2018), el patrullaje vehicular está enfocado en vigilar y recorrer sobre amplias áreas geográficas para brindar una atención rápida a la comunidad.

2.3.3. Herramientas digitales

2.3.3.1. Visual Studio 2019

Es un entorno de desarrollo para diseñar aplicaciones multiplataforma en dispositivos móviles y computadoras con el lenguaje .NET (Microsoft, 2022a).

2.3.3.2. Microsoft Excel

Es un programa informático que permite organizar datos en hojas de calculo para realizar plantillas con gráficos, tablas entre otras herramientas, además tiene formulas modernas para realizar diversos cálculos (Microsoft, 2022b).

2.3.3.3. Google Maps

Es un aplicativo que brinda información de lugares, rutas entre otros incidentes en tiempo real (Google, 2022).

2.3.3.4 SIPCOP

El Sistema Informático de Planificación y Control de Patrullaje Policial (SIPCOP), es un software del Ministerio Interior del Perú que permite monitorear mediante GPS, el desplazamiento de los vehículos policiales en tiempo real (Ministerio Interior, 2018).

2.4. Definición de términos

Jurisdicción comisaria: Territorio que está bajo la supervisión de una comisaria para ejercer sus funciones y competencias que la ley le otorga, con el objetivo de brindar seguridad al territorio en cuestión (MININTER & Sub Comandancia General PNP, 2018).

Sector: Es un área delimitada por la jurisdicción de una comisaría. El conjunto de sectores conforma el área de jurisdicción de una comisaria (MININTER & Sub Comandancia General PNP, 2018).

Subsector: Es un espacio geográfico que se encuentran dentro de un sector. Esta división aporta una adecuada asignación de recursos necesarios para el patrullaje policial (MININTER & Sub Comandancia General PNP, 2018).

Mapa de delito: Es la representación de un espacio geográfico determinado, que permite la formulación de estrategias de patrullaje dirigidas a sectores o subsectores que contengan mayor incidencia delictiva. La información contenida en el mapa son los tipos de delitos registrados en los cuadernos de denuncias de las comisarias (MININTER & Sub Comandancia General PNP, 2018).

Puntos críticos: Son espacios conocidos que amenazan la seguridad de la ciudadanía y son de interés en las operaciones policiales (MININTER & Sub Comandancia General PNP, 2018).

III. Hipótesis y Variables

3.1. Hipótesis

Hipótesis General

Las rutas diseñadas por metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en las jurisdicciones de las comisarías del distrito del Rímac.

Hipótesis Específicas

1) Las rutas diseñadas por metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la comisaría PNP Piedra Liza.
2) Las rutas diseñadas por metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la comisaría PNP Rímac.
3) Las rutas diseñadas por metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la comisaría PNP Manzano.

4) Las rutas diseñadas por metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la comisaría PNP Ciudad y Campo.
5) Las rutas diseñadas por la metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la comisaría PNP Flor de Amancaes.

3.2. Variables

Variable Independiente

Rutas diseñadas por metaheurísicas.

Variable Dependiente

Optimización de recorridos.

3.3. Operacionalización de variables

Tabla 2

Operacionalización de las variables

Variables	Definición Conceptual	Definición Operacional	Dimensiones	Indicadores	Escala de medición
Rutas diseñadas por metaheurísticas	Para el caso particular del problema de ruteo de vehículos, las metaheurísticas ejecutan procedimientos eficientes durante la exploración del espacio de soluciones para llegar a diseñar rutas casi optimas en un tiempo muy reducido (Reyes Rueda, 2020).	Consiste en usar metaheurísiticas para la construcción de rutas de patrullaje en el menor tiempo posible, considerando a los puntos críticos los destinos a visitar.	Optimización en la construcción de ruta	Tiempo	Numérica (segundos)
Optimización del recorrido	"La optimización de rutas es el proceso para decidir la ruta más eficiente en cuanto a costo" (Verizon Connect, 2022).	Consiste en determinar la ruta que permita al vehículo policial realizar desplazamientos cortos sobre los puntos críticos.	Optimización en el recorrido de puntos críticos	Distancia	Numérica (metros)

IV. Materiales y Métodos

4.1. Área de estudio

La presente investigación tiene como ámbito geográfico el distrito del Rímac; limita al norte con el distrito de Independencia; al este con el distrito de San Juan Lurigancho; al sur con el Cercado de Lima y al oeste con el distrito de San Martin de Porres. El área de estudio son las cinco jurisdicciones que pertenecen al distrito, siendo los siguientes:

- La jurisdicción de la CPNP Piedra Liza, está dividida en 2 sectores y cada uno de estas en 2 subsectores, por lo tanto, esta área de jurisdicción está divida en cuatro partes. Ver Anexo 2.
- La jurisdicción de la CPNP Rímac, esta dividida en 6 sectores. Ver Anexo 3.
- La jurisdicción de la CPNP Manzano, esta dividida en 6 sectores. Ver Anexo 4.
- La jurisdicción de la CPNP Ciudad y Campo, esta divida en 3 sectores y cada una de estas en 2 subsectores, por lo tanto, esta área de jurisdicción está divida en seis partes. Ver Anexo 5.
- La jurisdicción de la CPNP Flor de Amancaes, está divida en 2 sectores y cada uno de estas en 2 subsectores, por lo tanto, esta área de jurisdicción está divida en cuatro partes. Ver Anexo 6.

La planificación del patrullaje vehicular se realiza mediante el uso del mapa de delito y hoja de ruta, puesto que en estos documentos se detallan los lugares o destinos que deben ser visitados por los vehículos.

El mapa de delito brinda la ubicación referencial de lugares que han sido recurrentemente víctimas de delitos (puntos críticos), este mapa es actualizado mensualmente con los registros de denuncias. Esta información es usada para dirigir la actividad del patrullaje en visitar los puntos identificados en el mapa.

La hoja de ruta como se muestra en el Anexo 7, es una ficha que brinda información general sobre los horarios, caminos y localidades a considerar en la rutina del patrullaje según el sector asignado. Con respecto a estas localidades para este trabajo serán llamados puntos de protección, estos puntos pueden ser colegios, hospitales, parroquias entre otros. Otra información relevante en este documento son los puestos tácticos.

Según la hoja de ruta, el patrullaje se realiza en tres tipos de modalidades: táctico, móvil y ronda de jurisdicción. Estas actividades se ejecutan según un horario definido como se muestra en la Tabla 3.

Tabla 3

Programación de horarios de patrullaje

Horario	Patrullaje
06:00 a 07:00	Táctico
07:00 a 12:00	Móvil
12:00 a 13:00	Táctico
13:00 a 18:00	Móvil
18:00 a 19:00	Táctico
19:00 a 00:00	Móvil
00:00 a 06:00	Ronda Jurisdicción

Nota. Los horarios mostrados son comunes en todas las comisarias, sin embargo, puede variar por disposiciones de alguna eventualidad.

Patrullaje Táctico: Esta actividad consiste en que el vehículo policial debe estacionarse durante largos periodos de tiempo en un lugar estratégico.

Patrullaje Móvil: Esta actividad consiste en que el vehículo se encuentre en constante desplazamiento sobre el sector que le fue asignado.

Ronda Jurisdicción: Esta actividad consiste en que el vehículo durante el horario nocturno realice desplazamientos sobre porciones más pequeñas de un sector.

En la hoja de ruta está indicada, según sea el sector a patrullar, la siguiente información relevante:

Zona de Patrullaje: Indica las avenidas, calles, jirones, etc. que el vehículo debe usar como caminos durante su desplazamiento.

Puestos Tácticos: Son los lugares que el vehículo utiliza para detenerse durante el patrullaje táctico o para iniciar un nuevo recorrido. Esta información va cambiando semana a semana en la hoja de ruta.

Adicionalmente a los datos expuestos, también están indicados puntos de protección o lugares vulnerables, como: centros comerciales, parques, colegios, parroquias entre otros, que deben ser vigilados durante el patrullaje, estos puntos también son considerados a visitar por los vehículos policiales.

Las rutinas de patrullaje comienzan por lo general desde un puesto táctico y a partir de ese punto, los vehículos comienzan a realizar recorridos sobre las zonas de patrullaje hasta llegar a visitar a los puntos críticos (identificados en el mapa de delito) o puntos de protección (indicados en la hoja de ruta). En lo que respecta a la decisión de las rutas o trayectorias a elegir durante el patrullaje, este se realiza de manera empírica.

Una vez que el vehículo se encuentre en movimiento, desde la comisaria, otro oficial utiliza el software SIPCOP para monitorear en tiempo real los recorridos que realiza el móvil e indicarle si fuese el caso, que atienda una eventual emergencia.

La rutina de patrullaje de cada vehiculo es anotado en un “Cuaderno de rutas” , sin embargo, son anotaciones generales que no precisan las rutas que se hayan realizado.

La presente investigación está enfocada en la actividad de patrullaje vehicular de tipo móvil. Debido a que la información requerida es confidencial, este trabajo cuenta con las constancias de autorización firmado por las autoridades de cada comisaria, para presentar los datos con fines académicos. Los documentos se encuentran adjuntados en las hojas finales del presente trabajo.

4.2. Tipo y diseño de investigación

4.2.1. Tipo de investigación

El presente trabajo de investigación se considera como descriptivo y aplicado.

Las investigaciones descriptivas, “pretenden medir o recoger información de manera independiente o conjunta sobre los conceptos o las variables a las que se refieren, esto es, su objetivo no es indicar cómo se relacionan éstas” (Hernández, 2018, p. 80). De esta forma la presente investigación recolecta información sobre la metodología de patrullaje vehicular, para identificar las características que se debe de tomar en cuenta en el diseño rutas de patrullaje con técnicas de la Investigación Operativa.

La investigación aplicada, “tiene como propósito corroborar la teoría, de manera directa, en un campo concreto de aplicación”(Moreno, 1987, p. 37). De igual manera esta investigación revisó la teoría correspondiente al problema del ruteo de vehículos, heurísticas, metaheurísticas, entre otras, para aplicarlos a la solución de la problemática expuesta.

4.2.2. Diseño de investigación

La investigación tiene un diseño no experimental y transversal.

No experimental se define como “la investigación que se realiza sin manipular deliberadamente variables. Lo que hacemos es observar fenómenos tal y como se dan en su contexto natural, para después analizarlos”(Gómez, 2006, p. 45). La presente investigación sigue lo definido, porque luego de obtener los resultados de la ejecución del modelo propuesto, se describió y analizó cada uno de ellos sin la necesidad de emularlos en un entorno controlado.

Diseño transversal, “recolectan datos en un solo momento, en un tiempo único” (Gómez, 2006, p. 46). La presente investigación recopiló y analizó la información en un periodo de tiempo específico, por lo que es considerado una investigación con diseño transversal.

4.3. Población y Muestra

Población

La población es “un conjunto finito o infinito de elementos con características comunes para los cuales serán extensivas las conclusiones de la investigación. Ésta queda delimitada por el problema y por los objetivos del estudio”(Arias, 2012, p. 81). En esta investigación la población estuvo conformada por la información de la actividad del patrullaje en todas las modalidades ya sea en vehiculo ,en moto y a pie, del perido de tiempo de enero a junio del 2021 de las cinco comisarías del distrito del Rímac.

Muestra no probabilística

La muestra no probabilística es “un procedimiento de selección en el que se desconoce la probabilidad que tienen los elementos de la población para integrar la muestra”(Arias, 2012, p. 85). Para intereses de la investigación, en la muestra que se seleccionó, se desconoce la cantidad exacta de elementos que intervienen en la rutina de patrullaje vehicular, pero se consideró lo siguiente:

- Los puntos críticos que se identificaron en los mapas de delitos mensuales desde enero hasta junio del año 2021.
- Los puntos de protección y puestos tácticos que se identificaron en la hoja de ruta de la primera semana de junio del año 2021.

4.4. Procedimiento, Técnica e Instrumento de recolección

4.4.1. Técnica de recolección de datos y información

El Análisis documental, “es una técnica que sirve para obtener información que se encuentre en documentos (expedientes, historias clínicas, registros…) que forman parte del estudio como fuente de información” (Rios, 2017, p. 102). Se usó la técnica de Análisis documental para obtener información sobre los puntos críticos de tipo delitos contra el patrimonio y la información de la planificación del patrullaje. La elección de esta técnica se debe a que la información requerida se encuentran en registros policiales llamados mapa de delitos y las hojas de ruta respectivamente.

4.4.2. Instrumento de recolección de datos e información

La ficha de trabajo es “el instrumento que nos permite ordenar y clasificar los datos consultados, incluyendo nuestras observaciones y críticas, facilitando así la redacción del escrito” (Rodríguez Moguel, 2005, p. 97). Se hizó una ficha de trabajo para obtener las ubicaciones geográficas de los puntos críticos dentro del área de jurisdicción de cada comisaría, otra ficha de trabajo que se utilizó, es para el recojo de datos de los puntos de protección y puestos tácticos considerados en los desplazamientos del patrullaje. Los instrumentos se encuentran en Anexo 8.

4.4.3. Procedimientos de recolección de datos e información

Luego de recopilar los datos, esta es organizada en tablas donde se indica las coordenadas geográficas (latitud y longitud) y una descripción del lugar al que se hace referencia (Ver Anexo 9). Además, se debe tener en cuenta las siguientes notaciones:

- PC = Punto Crítico
- PP = Punto de Protección
- PT = Puesto Táctico

4.5. Análisis estadísticos

Se uso el programa Minitab para realizar análisis estadísticos que validen las hipótesis de la investigación.

V. Resultados

5.1. Método de diseño de rutas empíricas

En esta sección se presenta un método para construir rutas empíricas, debido a que la información de las trayectorias reales usadas en las rutinas de patrullaje no se encuentra disponibles dentro de una base de datos.

Los registros se realizan en un cuaderno de rutas correspondiente a cada vehículo policial, sin embargo, en este no se detalla con precisión la trayectoria o la secuencia de visitas que haya hecho cada vehículo.

Se va a considerar la construcción de rutas para solamente el primer turno de patrullaje móvil (07:00 hasta 12:00) y se van a plantear situaciones ideales que podrían darse durante la actividad del patrullaje, basados en los comentarios recibidos por parte de los oficiales durante la recolección de datos.

Cotidianamente los puntos críticos y puntos de protección son unificados para llamarse puntos críticos, sin embargo, para fines prácticos este concepto no será tomado en cuenta, permaneciendo cada tipo de puntos con su denominación.

La frecuencia de patrullaje durante el primer turno es equivalente a cinco horas continuas, en el mejor escenario la frecuencia seria de veinte recorridos. Por lo tanto, en una hora, cada vehículo debe realizar cuatro recorridos distintos. Se plantean cuatro escenarios para la construcción de rutas basados en el número de puntos a visitar, por lo que en cada escenario cada ruta visitará una cantidad diferente de destinos, siendo los siguientes:

- Escenario 1: 100% (Número total de puntos establecidos en el sector a patrullar).
- Escenario 2: 80% (Número total de puntos establecidos en el sector a patrullar).
- Escenario 3: 60% (Número total de puntos establecidos en el sector a patrullar).
- Escenario 4: 40% (Número total de puntos establecidos en el sector a patrullar).

Se va a tener en consideración que cada ruta a diseñar debe visitar como mínimo cuatro destinos, esta condición se debe a que visitar menos de lo indicado no reflejaría

un escenario real en la frecuencia de patrullaje. Mencionado lo anterior, si se presentase un escenario donde no se cumpla con la cantidad mínima de puntos a visitar, se dejará de lado este escenario y los posteriores a este.

La selección del punto de partida (puesto táctico) y de los destinos a visitar (punto crítico y puntos de protección) se va a realizar aleatoriamente con el apoyo de las fórmulas de Microsoft Excel. El siguiente ejemplo va a detallar la forma de selección aleatoria de los puntos mencionados. La tabla 4 muestra la información para el ejemplo sobre los puntos de partida y los destinos.

Tabla 4

Información de puntos

Índice	Punto Partida	Índice	Destinos
1	PT1	1	PC1
2	PT2	2	PC2
3	PT3	3	PC3
		4	PC4
		5	PC5
		6	PP1
		7	PP2
		8	PP3

Nota. Se asignó un índice numérico para cada punto.

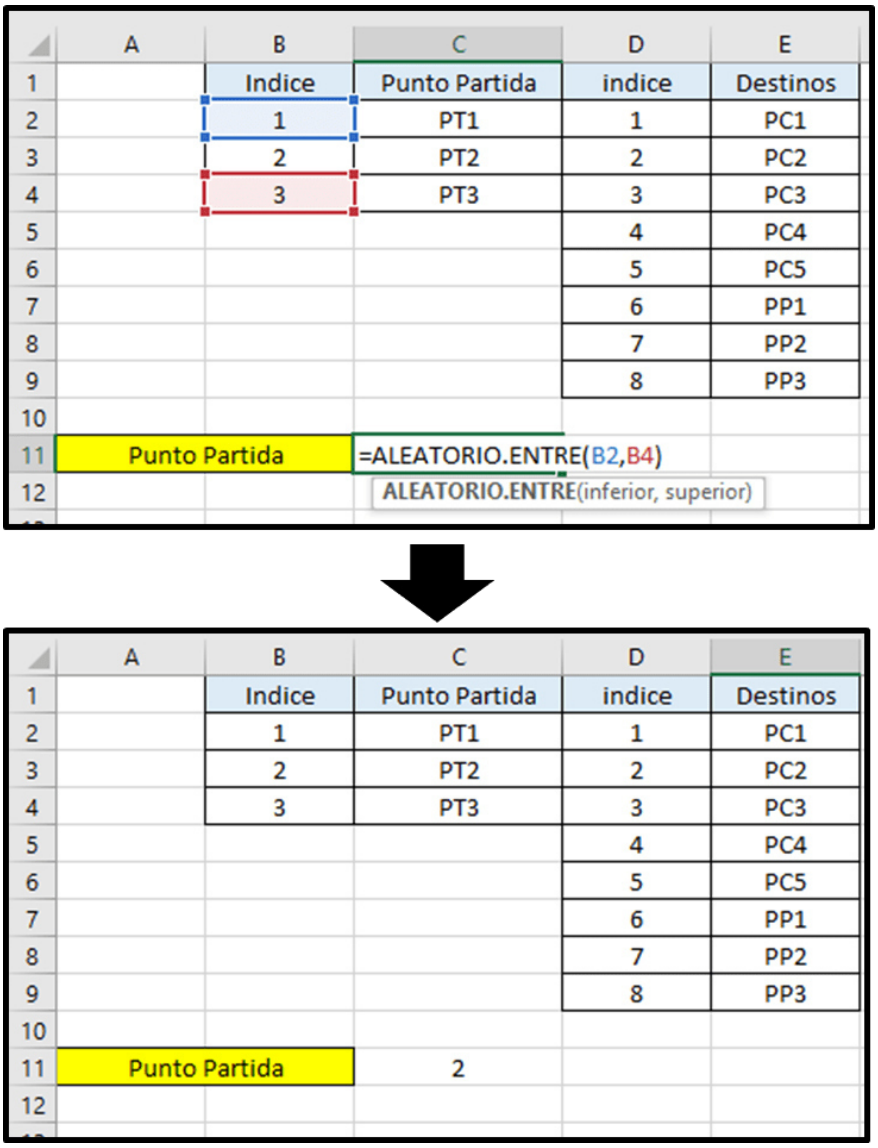
Los escenarios para el ejemplo son los siguientes:

- Escenario 1: 100% (8) = 8 puntos
- Escenario 2: 80% (8) = 6 puntos
- Escenario 3: 60% (8) = 5 puntos
- Escenario 4: 40% (8) = 3 puntos

El primer escenario considera visitar todos los puntos de un sector, en el ejemplo serian ocho destinos. En la selección aleatoria del punto de partida, se va a usar la siguiente fórmula de Excel: ***"= ALEATORIO. ENTRE (Inferior, Superior)"***. La Figura 5, muestra lo indicado.

Figura 5

Procedimiento de Selección aleatoria de Punto de Partida



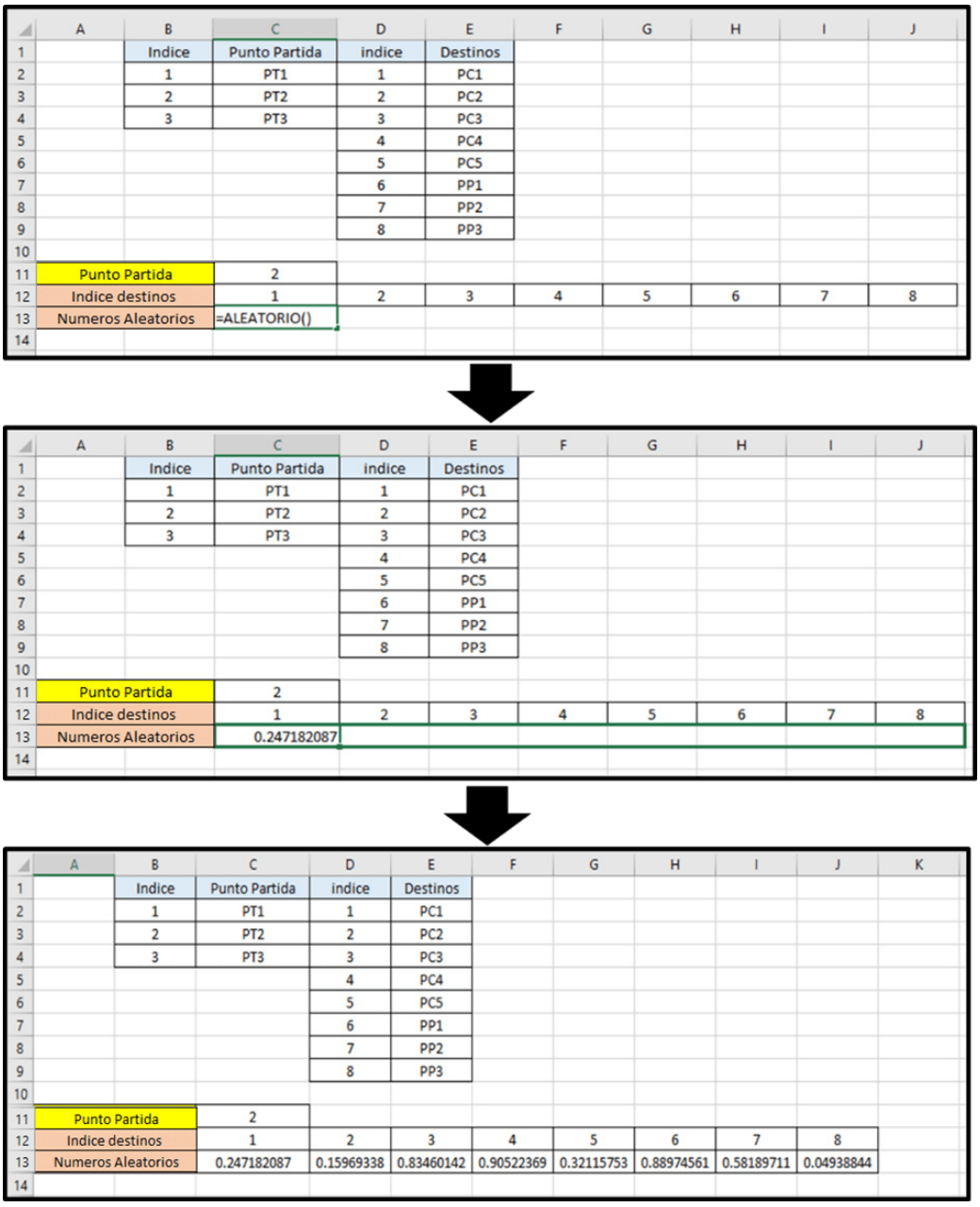
Nota. En la imagen se muestra los pasos de la función Aleatorio de Excel

De la Figura 5, el procedimiento es utilizar la función ***"= ALEATORIO. ENTRE (Inferior, Superior)"***, para seleccionar del conjunto de puntos de partida disponibles, el índice inferior y superior. Luego se identifica el índice que a sido seleccionado. Siendo para este ejemplo el Puesto Táctico 2 o PT2.

En la selección de los destinos a visitar secuencialmente, primero se van a crear número aleatorio entre 0 y 1 con la función ***"= ALEATORIO ( )"*** igual al total de destinos. En la Figura 6 se muestra lo descrito.

Figura 6

Procedimiento de creación de semillas



	Indice	Punto Partida	indice	Destinos
	1	PT1	1	PC1
	2	PT2	2	PC2
	3	PT3	3	PC3
			4	PC4
			5	PC5
			6	PP1
			7	PP2
			8	PP3

Punto Partida	2							
Indice destinos	1	2	3	4	5	6	7	8
Numeros Aleatorios	=ALEATORIO()							

	Indice	Punto Partida	indice	Destinos
	1	PT1	1	PC1
	2	PT2	2	PC2
	3	PT3	3	PC3
			4	PC4
			5	PC5
			6	PP1
			7	PP2
			8	PP3

Punto Partida	2							
Indice destinos	1	2	3	4	5	6	7	8
Numeros Aleatorios	0.247182087							

	Indice	Punto Partida	indice	Destinos
	1	PT1	1	PC1
	2	PT2	2	PC2
	3	PT3	3	PC3
			4	PC4
			5	PC5
			6	PP1
			7	PP2
			8	PP3

Punto Partida	2							
Indice destinos	1	2	3	4	5	6	7	8
Numeros Aleatorios	0.247182087	0.15969338	0.83460142	0.90522369	0.32115753	0.88974561	0.58189711	0.04938844

Nota. Se han creado ocho números aleatorios porque se presentan ocho destinos disponibles.

Luego de la creación de números aleatorios, se va usar la función **"= *JERARQUIA (número, referencia, [orden])*"** en los números creados.

Se debe tener en cuenta la siguiente información al momento de usar la función JERARQUIA:

- Número = Es el numero cuya jerarquía se desea conocer.
- Referencia = Es una lista de números. Esta lista debe contener al valor que se indica en “Número”.
- Orden = Es opcional. No se toma en consideración.

La función **“= *JERARQUIA (número, referencia, [orden])*“** tiene el siguiente procedimiento**:**

1. Se ordena descendentemente la lista de números que se indica en “Referencia”.
2. Se asigna una posición jerarquica a la lista de números que han sido ordenados descendentemente. El número más alto de la lista tendra la posición jerárquica equivalente al valor numérico de uno.
3. La funcion JERARQUIA va a buscar la posición jerárquica del valor que se indica en “Número”.

Para el ejemplo que se viene presentando, se utiliza la función JERARQUIA de la siguiente manera:

- Numero: Se va a seleccionar el primer número aleatorio obtenido.
- Referencia: Se va a seleccionar todos los números aleatorios.
- Orden: No se va a indicar nada.

De la tabla 6, si se utiliza la función JERARQUIA para el primer número aleatorio (0.247182087), el resultado obtenido seria la posicion jerarquica de 6. Se va a repetir este procedimiento según el escenario que se este desarrollando.

Para el primer escenario (visitar ocho puntos de ocho posibles), se elige aleatoriamente el punto de partida, luego se crean ocho números aleatorios entre 0 y 1 y se usa la función JERARQUIA a los ocho primeros números creados.

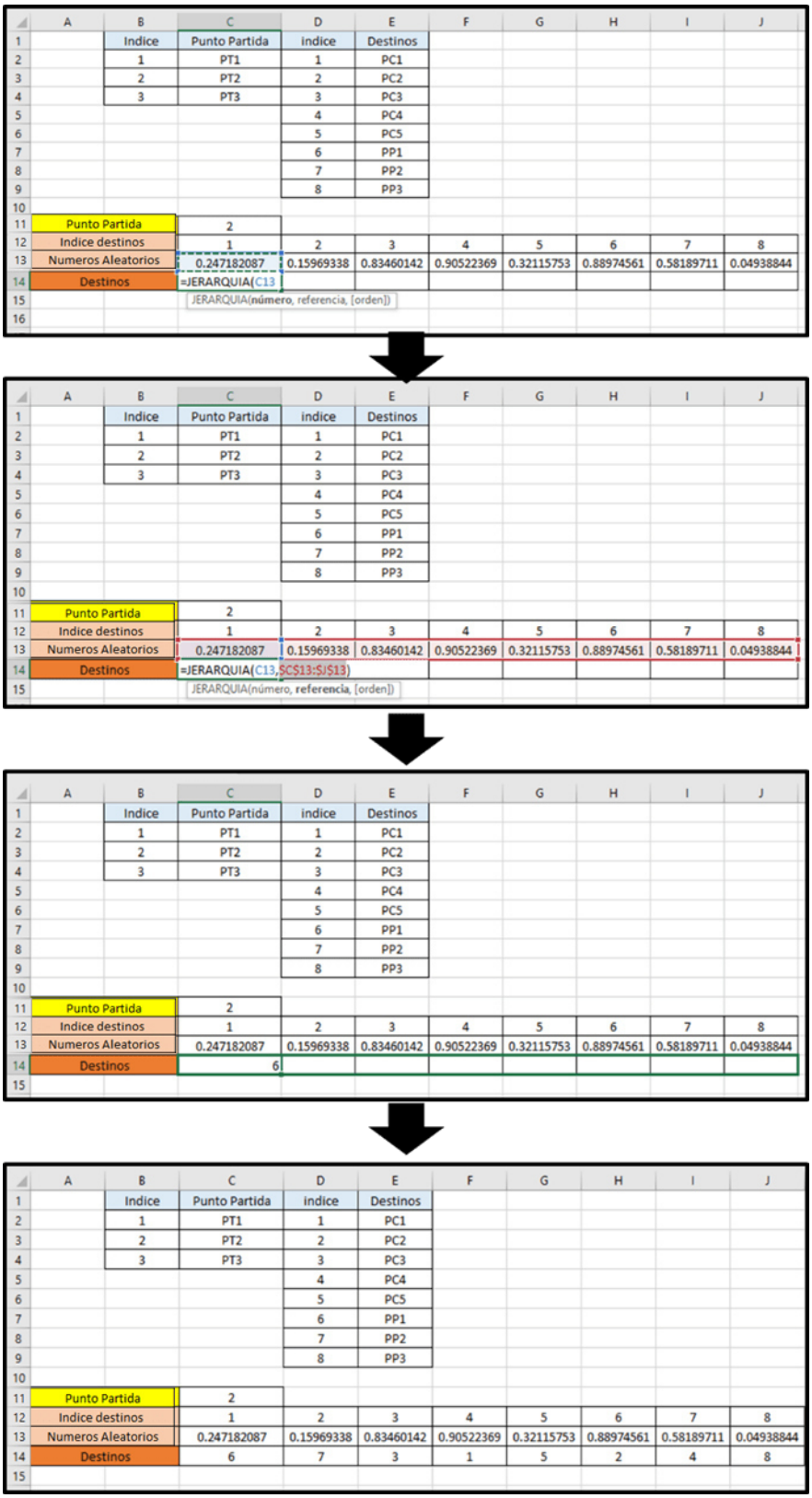
Para el segundo escenario (visitar seis puntos de ocho posibles), se elige aleatoriamente el punto de partida, luego se crean nuevamente ocho números aleatorios entre 0 y 1 y se usa la función JERARQUIA a los seis primeros números creados.

Para el tercer escenario (visitar cinco puntos de ocho posibles), se elige aleatoriamente el punto de partida, luego se crean nuevamente ocho números aleatorios entre 0 y 1 y se usa la función JERARQUIA a los cinco primeros números creados.

En la Figura 7 se muestra el procedimiento expuesto para el primer escenario .

Figura 7

Procedimiento de creación aleatoria de los destinos



Luego de la selección aleatoria de los destinos, se identifican la secuencia de las visitas según el índice. Siendo lo siguiente: PP1 – PP2 – PC3 – PC1 – PC5 – PC2 – PC4 –PP3 y adjuntando el punto de partida, la ruta empírica para el primer escenario sería: PT2 - PP1 – PP2 – PC3 – PC1 – PC5 – PC2 – PC4 – PP3 – PT2.

Para el segundo escenario, el procedimiento es considerar visitar 6 puntos. Se repite el mismo procedimiento expuesto. En la Figura 8 se muestra los resultados.

Figura 8

Resultados para el escenario 2

	A	B	C	D	E	F	G	H	I	J
1		Indice	Punto Partida	indice	Destinos					
2		1	PT1	1	PC1					
3		2	PT2	2	PC2					
4		3	PT3	3	PC3					
5				4	PC4					
6				5	PC5					
7				6	PP1					
8				7	PP2					
9				8	PP3					
10										
11	Punto Partida	3								
12	Indice destinos	1	2	3	4	5	6	7	8	
13	Numeros Aleatorios	0.794141051	0.49540552	0.03399507	0.63528285	0.67359044	0.9852644	0.05629325	0.96768864	
14	Destinos	3	6	8	5	4	1			
15										

De la Figura 8, la ruta para el segundo escenario es: PT3 – PC3 – PP1 – PP3 – PC5 – PC4 – PC1 – PT3. Así mismo en la Figura 9 se muestra los resultados para el tercer escenario.

Figura 9

Resultados para el escenario 3

	A	B	C	D	E	F	G	H	I	J
1		Indice	Punto Partida	indice	Destinos					
2		1	PT1	1	PC1					
3		2	PT2	2	PC2					
4		3	PT3	3	PC3					
5				4	PC4					
6				5	PC5					
7				6	PP1					
8				7	PP2					
9				8	PP3					
10										
11	Punto Partida	1								
12	Indice destinos	1	2	3	4	5	6	7	8	
13	Numeros Aleatorios	0.382736976	0.10097573	0.18910215	0.01649632	0.14983423	0.28587014	0.47932476	0.55445101	
14	Destinos	3	7	5	8	6				
15										

Según la Figura 9, la ruta es: PT1 – PC3 – PP2 –PC5–PP3–PP1–PT1. Finalmente, el cuarto escenario no se va a considerar, puesto que el número de puntos a visitar es menor a cuatro. La Tabla 5 muestra los resultados de las rutas empíricas construidas.

Tabla 5

Resultados finales de la creación de rutas empíricas

Escenarios	Ruta
1°	PT2 - PP1 – PP2 – PC3 – PC1 – PC5 – PC2 – PC4 – PP3 – PT2
2°	PT3 – PC3 – PP1 – PP3 – PC5 – PC4 – PC1 – PT3
3°	PT1 – PC3 – PP2 – PC5 – PP3 – PP1 – PT1

La razón para presentar este método de generación de rutas empíricas, es para compararlos con los resultados que se vayan a obtener con la aplicación de los algoritmos matemáticos.

5.2. Adaptación de algoritmos matemáticos para diseñar rutas de patrullaje

Se va a desarrollar mediante un ejemplo el diseño de las rutas para patrullaje vehicular aplicando heurísticas y metaheurísticas. La intención de presentar los procedimientos, es explicar detalladamente los pasos que cada algoritmo utiliza para su ejecución, los que se replicarán en el aplicativo desktop.

En primera instancia, se va a construir una matriz de distancias asociada a todos los puntos que van a ser considerados para el diseño de rutas. Se va a utilizar la fórmula Haversine o Semiverseno, esta es una ecuación que “calcula la distancia del círculo máximo entre dos puntos del globo sabiendo su longitud y latitud” (Torres Soler & Gárzon Torres, 2016, p. 81). La decisión de utilizar esta fórmula se debe a que, en contraposición a la distancia euclidiana, no se considera a la distancia como la longitud de una recta entre dos puntos. Rentería y Soto (2018) mencionan que utilizar este tipo de distancias representan con mas exactitud recorridos en zonas urbanas. La ecuación es la siguiente:

$$D = 2 * r * \arcsin\left[\sqrt{sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_2) * \cos(\phi_1) * sin^2\left(\frac{\Lambda_2 - \Lambda_1}{2}\right)}\right] \quad (2)$$

Donde:

- r es el radio de la tierra (en promedio 6371000 metros).
- ϕ_1 es la longitud del punto 1.
- ϕ_2 es la longitud del punto 2.

- Λ_1 es la latitud del punto 1.
- Λ_2 es la latitud del punto 2.

La información que se necesita para la Ecuación 2 son la latitud y longitud de cada punto en formato de radianes. Debido a que la información que se recopiló están en unidades sexagesimales, se hace la transformación modificando la Ecuación 2 de la siguiente manera:

$$D = 2 * r * \arcsin\left[\sqrt{sin^2\left(K * (\frac{\phi_2 - \phi_1}{2})\right) + \cos(K * \phi_2) * \cos(K * \phi_1) * sin^2\left(K * (\frac{\Lambda_2 - \Lambda_1}{2})\right)}\right] \quad (3)$$

Donde:

- $K = \frac{PI}{180}$

El resultado obtenido de la Ecuación 3 estará en metros. Se va a presentar el diseño de rutas con cada algoritmo para la información de la Tabla 6.

Tabla 6

Puntos seleccionados

Puntos	Coordenadas Geográficas	
	X	y
PT2	-12.0307752	-77.0437762
PC1	-12.0319187	-77.0435151
PC2	-12.0313547	-77.0430612
PC3	-12.0318362	-77.0426249
PC4	-12.0324533	-77.041923
PC5	-12.0337729	-77.0414756
PP1	-12.0338333	-77.0403727
PP2	-12.0342553	-77.0419098
PP3	-12.0356606	-77.0409329

Nota. Los puntos seleccionados son del ejemplo del primer escenario presentados en la sección anterior.

Se utiliza la información de la Tabla 6 y la Ecuación 3 en la construcción de la matriz de distancia. La Tabla 7 muestra lo expuesto.

Tabla 7

Matriz de distancias para el ejemplo

	PT2	PC1	PC2	PC3	PC4	PC5	PP1	PP2	PP3
PT2	0	128	254	411	661	837	1213	707	1064
PC1	128	0	165	311	558	727	1108	590	948
PC2	254	165	0	157	407	585	959	462	816
PC3	411	311	157	0	250	429	802	314	663
PC4	661	558	407	250	0	187	552	141	427
PC5	837	727	585	429	187	0	385	156	240
PP1	1213	1108	959	802	552	385	0	538	242
PP2	707	590	462	314	141	156	538	0	359
PP3	1064	948	816	663	427	240	242	359	0

Nota. En la primera fila y columna se va a ubicar el punto de partida. Distancia en metros.

Durante la aplicación de las técnicas matemáticas, se va a utilizar la matriz construida en los cuatro algoritmos.

Las heurísticas expuestas en las bases teóricas de esta investigación están enfocadas en resolver el CVRP, esta elección se debe a que en sus procedimientos se considera la restricción de capacidad del vehículo, esto implica que se genere un conjunto de rutas. Sin embargo, para fines de la investigación, esta restricción fue cambiada.

La condición de capacidad del vehículo se cambió por el número máximo de puntos o destinos que van a ser incluidos en una ruta. Para el caso de la Tabla 6 se busca construir una sola trayectoria que permita visitar a todos los puntos, por lo tanto, la condición es ocho puntos como máximo a insertar en la ruta.

Diseño de rutas con el algoritmo de Clarke y Wright

El algoritmo de Clarke y Wright es utilizado para problemas en los que se tiene n clientes, cada uno con demanda conocida y estos son visitados por vehículos que parten y retornan del depósito inicial. El objetivo del algoritmo es buscar rutas que minimicen la distancia de recorrido y al mismo tiempo satisfacer la demanda, teniendo en cuenta que no supere la capacidad de los vehículos.

Se han tomado en cuenta otros factores a los descritos en el párrafo anterior para que se ajuste a la situación problemática en estudio. Los clientes a visitar serán los

puntos críticos o puntos de protección y la restricción de capacidad del vehículo es cambiada por el número máximo de puntos que puede incluirse en una ruta de patrullaje.

Los elementos considerados serian:

- Punto críticos: $PC_i,\ i = 1, 2, \dots, n$
- Punto protección: $PP_i,\ i = 1, 2, \dots, n$
- Distancia entre puntos : d_{ij}

Para el ejemplo la condición es que la ruta contenga como máximo ocho puntos.

Luego de los cambios mencionados, se sigue los pasos del algoritmo de Clarke y Wright versión secuencial que han sido expuestos en las bases teóricas. Salinero (2016) detalla los procedimientos del algoritmo mediante un ejemplo, se va a utilizar el procedimiento expuesto para una mejor compresión del algoritmo.

La matriz de ahorro para la Tabla 7 es calculada aplicando la Ecuación 1 o con la fórmula del ahorro. La tabla 8 muestra la matriz de ahorro.

Tabla 8

Matriz de ahorro de distancias para el ejemplo

	PC1	PC2	PC3	PC4	PC5	PP1	PP2	PP3
PC1	0	217	228	231	238	233	245	244
PC2	217	0	508	508	506	508	499	502
PC3	228	508	0	822	819	822	804	812
PC4	231	508	822	0	1311	1322	1227	1298
PC5	238	506	819	1311	0	1665	1388	1661
PP1	233	508	822	1322	1665	0	1382	2035
PP2	245	499	804	1227	1388	1382	0	1412
PP3	244	502	812	1298	1661	2035	1412	0

De la Tabla 8 se va a considerar que las distancias son simétricas de ida y vuelta entre dos puntos. A esta matriz se le va a añadir una columna y fila para que represente el punto de partida o el PT2.

Tabla 8.1

Matriz de ahorro de distancias para el ejemplo

	PC1	PC2	PC3	PC4	PC5	PP1	PP2	PP3	PT2
PC1	0	217	228	231	238	233	245	244	0
PC2	217	0	508	508	506	508	499	502	0
PC3	228	508	0	822	819	822	804	812	0
PC4	231	508	822	0	1311	1322	1227	1298	0
PC5	238	506	819	1311	0	1665	1388	1661	0
PP1	233	508	822	1322	1665	0	1382	2035	0
PP2	245	499	804	1227	1388	1382	0	1412	0
PP3	244	502	812	1298	1661	2035	1412	0	0
PT2	0	0	0	0	0	0	0	0	0

El primer paso es seleccionar el ahorro más alto de toda la matriz. De la tabla 8.1, el elemento más alto es el desplazamiento de PP3 a PP1 o al revés. La ruta formada hasta el momento sería:

- Ruta: PT2 – PP3 – PP1.
- Puntos incluidos: 2 puntos.

Para evitar que se vuelva a escoger el punto de partida del tramo añadido (PP3), se va a penalizar en la matriz de la Tabla 8.1. la fila y columna colocando “-1” en la posición de PP3. La Tabla 8.2. muestra lo indicado.

Tabla 8.2

Matriz de ahorro de distancias para el ejemplo

	PC1	PC2	PC3	PC4	PC5	PP1	PP2	PP3	PT2
PC1	0	217	228	231	238	233	245	-1	0
PC2	217	0	508	508	506	508	499	-1	0
PC3	228	508	0	822	819	822	804	-1	0
PC4	231	508	822	0	1311	1322	1227	-1	0
PC5	238	506	819	1311	0	1665	1388	-1	0
PP1	233	508	822	1322	1665	0	1382	-1	0
PP2	245	499	804	1227	1388	1382	0	-1	0
PP3	-1	-1	-1	-1	-1	-1	-1	-1	0
PT2	0	0	0	0	0	0	0	0	0

El siguiente paso es seleccionar de la fila donde se encuentra el punto de llegada del tramo añadido (PP1), el mayor de los ahorros y se identifica el punto de la columna asociada. El procedimiento se ve en la Tabla 8.3.

Tabla 8.3

Matriz de ahorro de distancias para el ejemplo

	PC1	PC2	PC3	PC4	PC5	PP1	PP2	PP3	PT2
PP1	233	508	822	1322	1665	0	1382	-1	0

El elemento más alto es el desplazamiento de PP1 a PC5. La ruta formada hasta el momento sería:

- Ruta: PT2 – PP3 – PP1 – PC5.
- Puntos incluidos: 3 puntos.

Como en el procedimiento anterior, se va a penalizar en la matriz de la Tabla 8.2. la fila y columna del primer punto correspondiente al último tramo de ruta añadido (PP1), colocando “-1” en la posición indicada. La Tabla 8.4. muestra lo indicado.

Tabla 8.4

Matriz de ahorro de distancias para del ejemplo

	PC1	PC2	PC3	PC4	PC5	PP1	PP2	PP3	PT2
PC1	0	217	228	231	238	-1	245	-1	0
PC2	217	0	508	508	506	-1	499	-1	0
PC3	228	508	0	822	819	-1	804	-1	0
PC4	231	508	822	0	1311	-1	1227	-1	0
PC5	238	506	819	1311	0	-1	1388	-1	0
PP1	-1	-1	-1	-1	-1	-1	-1	-1	0
PP2	245	499	804	1227	1388	-1	0	-1	0
PP3	-1	-1	-1	-1	-1	-1	-1	-1	0
PT2	0	0	0	0	0	0	0	0	0

Nuevamente el siguiente paso es seleccionar de la fila donde se encuentra el punto de llegada del último tramo añadido (PC5), el mayor de los ahorros y se identifica el punto de la columna asociada. El procedimiento se ve en la Tabla 8.5.

Tabla 8.5

Matriz de ahorro de distancias para el ejemplo

	PC1	PC2	PC3	PC4	PC5	PP1	PP2	PP3	PT2
PC5	238	506	819	1311	0	-1	1388	-1	0

El elemento más alto es el desplazamiento de PC5 a PP2. La ruta formada hasta el momento sería:

- Ruta: PT2 – PP3 – PP1 – PC5 – PP2.
- Puntos incluidos: 4 puntos.

Se va a penalizar en la matriz de la Tabla 8.4. la fila y columna del primer punto del último tramo de ruta añadido (PC5), colocando "-1" en la posición indicada. La Tabla 8.6. muestra lo indicado.

Tabla 8.6

Matriz de ahorro de distancias para del ejemplo

	PC1	PC2	PC3	PC4	PC5	PP1	PP2	PP3	PT2
PC1	0	217	228	231	-1	-1	245	-1	0
PC2	217	0	508	508	-1	-1	499	-1	0
PC3	228	508	0	822	-1	-1	804	-1	0
PC4	231	508	822	0	-1	-1	1227	-1	0
PC5	-1	-1	-1	-1	-1	-1	-1	-1	0
PP1	-1	-1	-1	-1	-1	-1	-1	-1	0
PP2	245	499	804	1227	-1	-1	0	-1	0
PP3	-1	-1	-1	-1	-1	-1	-1	-1	0
PT2	0	0	0	0	0	0	0	0	0

Se repite el paso anterior, seleccionar de la fila donde se encuentra el punto de llegada del último tramo añadido (PP2), el mayor de los ahorros y se identifica el punto de la columna asociada. El procedimiento se ve en la Tabla 8.7.

Tabla 8.7

Matriz de ahorro de distancias para del ejemplo

	PC1	PC2	PC3	PC4	PC5	PP1	PP2	PP3	PT2
PP2	245	499	804	1227	-1	-1	0	-1	0

El elemento más alto es el desplazamiento de PP2 a PC4. La ruta formada hasta el momento sería:

- Ruta: PT2 – PP3 – PP1 – PC5 – PP2 – PC4.
- Puntos incluidos: 5 puntos.

Se va a penalizar en la matriz de la Tabla 8.6. la fila y columna del primer punto del último tramo de ruta añadido (PP2), colocando "-1" en la posición indicada. La Tabla 8.8. muestra lo indicado.

Tabla 8.8

Matriz de ahorro de distancias para el ejemplo

	PC1	PC2	PC3	PC4	PC5	PP1	PP2	PP3	PT2
PC1	0	217	228	231	-1	-1	-1	-1	0
PC2	217	0	508	508	-1	-1	-1	-1	0
PC3	228	508	0	822	-1	-1	-1	-1	0
PC4	231	508	822	0	-1	-1	-1	-1	0
PC5	-1	-1	-1	-1	-1	-1	-1	-1	0
PP1	-1	-1	-1	-1	-1	-1	-1	-1	0
PP2	-1	-1	-1	-1	-1	-1	-1	-1	0
PP3	-1	-1	-1	-1	-1	-1	-1	-1	0
PT2	0	0	0	0	0	0	0	0	0

Se vuelve a seleccionar de la fila donde se encuentra el punto de llegada del último tramo añadido (PC4), el mayor de los ahorros y se identifica el punto de la columna asociada. El procedimiento se ve en la Tabla 8.9.

Tabla 8.9

Matriz de ahorro de distancias para del ejemplo

	PC1	PC2	PC3	PC4	PC5	PP1	PP2	PP3	PT2
PC4	231	508	822	0	-1	-1	-1	-1	0

El elemento más alto es el desplazamiento de PC4 a PC3. La ruta formada hasta el momento sería:

- Ruta: PT2 – PP3 – PP1 – PC5 – PP2 – PC4 – PC3.
- Puntos incluidos: 6 puntos.

Se va a penalizar en la matriz de la Tabla 8.8. la fila y columna del primer punto del último tramo de ruta añadido (PC4), colocando “-1” en la posición indicada. La Tabla 8.10. muestra lo indicado.

Tabla 8.10

Matriz de ahorro de distancias para el ejemplo

	PC1	PC2	PC3	PC4	PC5	PP1	PP2	PP3	PT2
PC1	0	217	228	-1	-1	-1	-1	-1	0
PC2	217	0	508	-1	-1	-1	-1	-1	0
PC3	228	508	0	-1	-1	-1	-1	-1	0
PC4	-1	-1	-1	-1	-1	-1	-1	-1	0
PC5	-1	-1	-1	-1	-1	-1	-1	-1	0
PP1	-1	-1	-1	-1	-1	-1	-1	-1	0
PP2	-1	-1	-1	-1	-1	-1	-1	-1	0
PP3	-1	-1	-1	-1	-1	-1	-1	-1	0
PT2	0	0	0	0	0	0	0	0	0

Nuevamente el siguiente paso es seleccionar de la fila donde se encuentra el punto de llegada del último tramo añadido (PC3), el mayor de los ahorros y se identifica el punto de la columna asociada. El procedimiento se ve en la Tabla 8.11.

Tabla 8.11

Matriz de ahorro de distancias para el ejemplo

	PC1	PC2	PC3	PC4	PC5	PP1	PP2	PP3	PT2
PC3	228	508	0	-1	-1	-1	-1	-1	0

El elemento más alto es el desplazamiento de PC3 a PC2. La ruta formada hasta el momento sería:

- Ruta: PT2 – PP3 – PP1 – PC5 – PP2 – PC4 – PC3 – PC2.
- Puntos incluidos: 7 puntos.

Se va a penalizar en la matriz de la Tabla 8.10 la fila y columna del primer punto del último tramo de ruta añadido (PC4), colocando “-1” en la posición indicada. La Tabla 8.12. muestra lo indicado.

Tabla 8.12

Matriz de ahorro de distancias para el ejemplo

	PC1	PC2	PC3	PC4	PC5	PP1	PP2	PP3	PT2
PC1	0	217	-1	-1	-1	-1	-1	-1	0
PC2	217	0	-1	-1	-1	-1	-1	-1	0
PC3	-1	-1	-1	-1	-1	-1	-1	-1	0
PC4	-1	-1	-1	-1	-1	-1	-1	-1	0
PC5	-1	-1	-1	-1	-1	-1	-1	-1	0
PP1	-1	-1	-1	-1	-1	-1	-1	-1	0
PP2	-1	-1	-1	-1	-1	-1	-1	-1	0
PP3	-1	-1	-1	-1	-1	-1	-1	-1	0
PT2	0	0	0	0	0	0	0	0	0

El siguiente paso es seleccionar de la fila donde se encuentra el punto de llegada del último tramo añadido (PC2), el mayor de los ahorros y se identifica el punto de la columna asociada. El procedimiento se ve en la Tabla 8.13.

Tabla 8.13

Matriz de ahorro de distancias para el ejemplo

	PC1	PC2	PC3	PC4	PC5	PP1	PP2	PP3	PT2
PC2	217	0	-1	-1	-1	-1	-1	-1	0

El elemento más alto es el desplazamiento de PC2 a PC1. La ruta formada hasta el momento sería:

- Ruta: PT2 – PP3 – PP1 – PC5 – PP2 – PC4 – PC3 – PC2 – PC1.
- Puntos incluidos: 8 puntos.

Según este último resultado ya se han insertado el máximo número de destinos (8 puntos), por lo que se va a penalizar en la matriz de la Tabla 8.12 la fila y columna de los puntos del último tramo añadido (PC2 y PC1), colocando “-1” en la posición indicada. La Tabla 8.14. muestra lo indicado.

Tabla 8.14

Matriz de ahorro de distancias para el ejemplo

	PC1	PC2	PC3	PC4	PC5	PP1	PP2	PP3	PT2
PC1	-1	-1	-1	-1	-1	-1	-1	-1	0
PC2	-1	-1	-1	-1	-1	-1	-1	-1	0
PC3	-1	-1	-1	-1	-1	-1	-1	-1	0
PC4	-1	-1	-1	-1	-1	-1	-1	-1	0
PC5	-1	-1	-1	-1	-1	-1	-1	-1	0
PP1	-1	-1	-1	-1	-1	-1	-1	-1	0
PP2	-1	-1	-1	-1	-1	-1	-1	-1	0
PP3	-1	-1	-1	-1	-1	-1	-1	-1	0
PT2	0	0	0	0	0	0	0	0	0

El resultado final de la aplicación del algoritmo y asociándole la distancia según la matriz de la Tabla 7 es:

- Ruta: PT2 – PP3 – PP1 – PC5 – PP2 – PC4 – PC3 – PC2 – PC1 – PT2.
- Distancia: 2688 metros.

En el caso que la condición haya sido diseñar rutas con un máximo de destinos menor a ocho, esto implicaría que no todos los destinos sean incluidos en una sola ruta, por ejemplo, si el caso hubiera sido incluir como máximo cuatro destinos, el resultado hubiera sido el siguiente:

- PT2 – PP3 – PP1 – PC5 – PP2 – PT2.

Una vez penalizado con “-1” la fila y columna de PC5 y PP2, se empieza a construir la siguiente ruta que contenga los destinos faltantes, el procedimiento es idéntico, se volvería a comenzar por seleccionar el valor más alto de toda la matriz teniendo en cuenta todas las penalizaciones hechas hasta ese momento.

El algoritmo termina cuando todos los destinos estén incluidos en alguna ruta.

Diseño de rutas con el algoritmo del Vecino más Cercano

El algoritmo del vecino más cercano consiste en ir generando rutas mediante la unión de puntos más cercanos entre sí, además como se indicó en las bases teóricas su procedimiento está basado en resolver el CVRP, por ello durante la construcción de las rutas se debe tener en cuenta que la demanda de los destinos que son incluidos en cada ruta no debe superar la capacidad del vehículo, sin embargo, al igual que en el caso del algoritmo de Clarke y Wright, para esta investigación la demanda de los destinos no

existe, por lo que la restricción de capacidad del vehículo es cambiado por el número máximo de puntos que puede incluirse en una ruta de patrullaje.

El trabajo de investigación de Ramírez (2016), consistió en aplicar el algoritmo del vecino más cercano para construir rutas secuencialmente teniendo como condición que en cada trayectoria no exceda un tiempo de desplazamiento. Para la presente investigación se va a realizar el mismo procedimiento hecho por el autor mencionado, pero cambiando la condición por el número máximo de puntos a visitar. En el ejemplo que se desarrolla hasta ahora son ocho destinos.

El primer paso consiste en fijar la fila del punto de partida (PT2) en la matriz de distancia de la Tabla 7, luego seleccionar el punto que genere el menor desplazamiento o la columna con el menor elemento de la fila de PT2. En la Tabla 9 se ve lo mencionado.

Tabla 9

Procedimiento del algoritmo del Vecino más Cercano

	PT2	PC1	PC2	PC3	PC4	PC5	PP1	PP2	PP3
PT2	0	128	254	411	661	837	1213	707	1064

Nota. No se considera la columna del punto de partida para la selección del menor elemento de la fila.

Según la Tabla 9, la ruta formada seria:

- Ruta: PT2 – PC1.
- Puntos incluidos: 1 punto.

Por ser el primer punto a visitar, se debe eliminar de la matriz de distancia la columna del último punto añadido a la ruta (PC1) y se realiza la búsqueda del menor desplazamiento en la fila del último punto añadido (PC1). En la Tabla 9.1 se ve lo mencionado.

Tabla 9.1

Procedimiento del algoritmo del Vecino más Cercano

	PT2	PC2	PC3	PC4	PC5	PP1	PP2	PP3
PT2	0	254	411	661	837	1213	707	1064
PC1	128	165	311	558	727	1108	590	948
PC2	254	0	157	407	585	959	462	816
PC3	411	157	0	250	429	802	314	663
PC4	661	407	250	0	187	552	141	427
PC5	837	585	429	187	0	385	156	240
PP1	1213	959	802	552	385	0	538	242
PP2	707	462	314	141	156	538	0	359
PP3	1064	816	663	427	240	242	359	0

Nota. No se considera la columna del punto de partida para la selección del menor elemento de la fila.

Según la Tabla 9.1, la ruta formada seria:

- Ruta: PT2 – PC1 – PC2.
- Puntos incluidos: 2 puntos.

Lo siguiente es eliminar en la matriz de distancia de la Tabla 9.1, la fila del primer punto del tramo añadido a la ruta (PC1) y la columna del último punto del tramo añadido a la ruta (PC2), luego se realiza la búsqueda del menor desplazamiento en la fila del último punto mencionado (PC2). En la Tabla 9.1 se ve lo indicado.

Tabla 9.2

Procedimiento del algoritmo del Vecino más Cercano

	PT2	PC3	PC4	PC5	PP1	PP2	PP3
PT2	0	411	661	837	1213	707	1064
PC2	254	157	407	585	959	462	816
PC3	411	0	250	429	802	314	663
PC4	661	250	0	187	552	141	427
PC5	837	429	187	0	385	156	240
PP1	1213	802	552	385	0	538	242
PP2	707	314	141	156	538	0	359
PP3	1064	663	427	240	242	359	0

Nota. No se considera la columna del punto de partida para la selección del menor elemento de la fila.

Según la Tabla 9.2, la ruta formada seria:

- Ruta: PT2 – PC1 – PC2 – PC3.
- Puntos incluidos: 3 puntos.

Nuevamente se elimina en la matriz de la Tabla 9.2, la fila del primer punto del tramo añadido a la ruta (PC2) y la columna del último punto del tramo añadido a la ruta (PC3), luego se realiza la búsqueda del menor desplazamiento en la fila del último punto mencionado (PC3). Ver la Tabla 9.3.

Tabla 9.3

Procedimiento del algoritmo del Vecino más Cercano

	PT2	PC4	PC5	PP1	PP2	PP3
PT2	0	661	837	1213	707	1064
PC3	411	250	429	802	314	663
PC4	661	0	187	552	141	427
PC5	837	187	0	385	156	240
PP1	1213	552	385	0	538	242
PP2	707	141	156	538	0	359
PP3	1064	427	240	242	359	0

Nota. No se considera la columna del punto de partida para la selección del menor elemento de la fila.

Según la Tabla 9.3, la ruta formada seria:

- Ruta: PT2 – PC1 – PC2 – PC3 – PC4.
- Puntos incluidos: 4 puntos.

Se vuelve a repetir el mismo paso en eliminar de la matriz de distancia de la Tabla 9.3, la fila del primer punto del tramo añadido a la ruta (PC3) y la columna del último punto del tramo añadido a la ruta (PC4), luego se realiza la búsqueda del menor desplazamiento en la fila del último punto mencionado (PC4). Ver la Tabla 9.4.

Tabla 9.4

Procedimiento del algoritmo del Vecino más Cercano

	PT2	PC5	PP1	PP2	PP3
PT2	0	837	1213	707	1064
PC4	661	187	552	141	427
PC5	837	0	385	156	240
PP1	1213	385	0	538	242
PP2	707	156	538	0	359
PP3	1064	240	242	359	0

Nota. No se considera la columna del punto de partida para la selección del menor elemento de la fila.

Según la Tabla 9.4, la ruta formada seria:

- Ruta: PT2 – PC1 – PC2 – PC3 – PC4 – PP2.
- Puntos incluidos: 5 puntos.

Se vuelve a repetir el mismo paso en eliminar de la matriz de distancia de la Tabla 9.4, la fila del primer punto del tramo añadido a la ruta (PC4) y la columna del último punto del tramo añadido a la ruta (PP2), luego se realiza la búsqueda del menor desplazamiento en la fila del último punto mencionado (PP2). En la Tabla 9.5 se ve lo indicado.

Tabla 9.5

Procedimiento del algoritmo del Vecino más Cercano

	PT2	PC5	PP1	PP3
PT2	0	837	1213	1064
PC5	837	0	385	240
PP1	1213	385	0	242
PP2	707	156	538	359
PP3	1064	240	242	0

Nota. No se considera la columna del punto de partida para la selección del menor elemento de la fila.

Según la Tabla 9.5, la ruta formada seria:

- Ruta: PT2 – PC1 – PC2 – PC3 – PC4 – PP2 – PC5.
- Puntos incluidos: 6 puntos.

Nuevamente se debe eliminar de la matriz de distancia de la Tabla 9.5, la fila del primer punto del tramo añadido a la ruta (PP2) y la columna del último punto del tramo añadido a la ruta (PC5), luego se realiza la búsqueda del menor desplazamiento en la fila del último punto mencionado (PC5). En la Tabla 9.6 se ve lo indicado.

Tabla 9.6

Procedimiento del algoritmo del Vecino más Cercano

	PT2	PP1	PP3
PT2	0	1213	1064
PC5	837	385	240
PP1	1213	0	242
PP3	1064	242	0

Nota. No se considera la columna del punto de partida para la selección del menor elemento de la fila.

Según la Tabla 9.6, la ruta formada seria:

- Ruta: PT2 – PC1 – PC2 – PC3 – PC4 – PP2 – PC5 – PP3.
- Puntos incluidos: 7 puntos.

Se debe eliminar en la matriz de distancia de la Tabla 9.6, la fila del primer punto del tramo añadido a la ruta (PC5) y la columna del último punto del tramo añadido a la ruta (PP3), luego se realiza la búsqueda del menor desplazamiento en la fila del último punto mencionado (PP3). En la Tabla 9.7 se ve lo indicado.

Tabla 9.7

Procedimiento del algoritmo del Vecino más Cercano

	PT2	PP1
PT2	0	1213
PP1	1213	0
PP3	1064	242

Nota. No se considera la columna del punto de partida para la selección del menor elemento de la fila.

Según la Tabla 9.7, la ruta formada seria:

- Ruta: PT2 – PC1 – PC2 – PC3 – PC4 – PP2 – PC5 – PP3 – PP1.
- Puntos incluidos: 8 puntos.

Según este último resultado ya se ha insertado el máximo número de destinos (ocho puntos), por lo que se va a eliminar de la Tabla 9.7, la columna del último punto del tramo añadido a la ruta (PP1).

El resultado final de la aplicación del algoritmo y asociándole la distancia según la matriz de la Tabla 7 es:

- Ruta: PT2 – PC1 – PC2 – PC3 – PC4 – PP2 – PC5 – PP3 – PP1 – PT2.
- Distancia: 2692 metros

En el caso que la condición haya sido diseñar rutas con un máximo de destinos menor a ocho, esto implicaría que no todos los destinos sean incluidos en una sola ruta, por ejemplo, si hubiera sido incluir como máximo cuatro destinos, el resultado hubiera sido el siguiente:

- Ruta: PT2 – PC1 – PC2 – PC3 – PC4 – PT2.

Una vez eliminada la columna de PC4, se empieza a construir la siguiente ruta que contenga los destinos faltantes, el procedimiento es idéntico, se volvería a comenzar por seleccionar de la fila del punto de partida o PT2, el punto que genere el menor desplazamiento o la columna del menor elemento de la fila de PT2 para luego continuar con los mismos pasos explicados.

El algoritmo termina cuando todos los destinos estén incluidos en alguna ruta.

Las metaheurísticas expuestas en las bases teóricas de esta investigación están enfocadas en resolver el TSP, esta elección se debe a que una vez encontrada las soluciones por las técnicas heurísticas, estas definen el punto de partida para cada ruta, además de los destinos a visitar en un solo recorrido. La descripción mencionada es semejante a la descripción del TSP, por lo que los algoritmos metaheurísticos elegidos usan procedimientos únicos para mejorar las soluciones obtenidas hasta el momento.

Diseño de rutas con el algoritmo de la Búsqueda Tabú

Para cada solución o ruta que haya sido obtenido del algoritmo del vecino más cercano se le va aplicar el algoritmo de la búsqueda tabú con el fin de mejorar los resultados. Sin embargo, habrá excepciones, si la alguna solución del algoritmo del vecino más cercano consiste en visitar menos de tres destinos, a este no se le va a aplicar el algoritmo de la búsqueda tabú.

Otras características que se van a considerar en la ejecución del TS son:

- El periodo de tenencia de un subrecorrido en la lista tabú será de dos iteraciones.
- La condición de parada del algoritmo será cuando se hayan hecho quince iteraciones.

Una vez indicadas estas particularidades, se sigue el mismo proceso expuesto en las bases teóricas. Para la presente investigación se va a realizar el mismo procedimiento hecho por Taha (2012) en el ejemplo que desarrollo en su libro (Ejemplo 11.5-1).

Se va a comenzar por asignarle un índice a cada punto involucrado en cada ruta. Esto para facilidad en las operaciones del algoritmo. En la Tabla 10 se muestra lo indicado.

Tabla 10

Índices de los destinos para el Algoritmo de la Búsqueda Tabú

Puntos	Índice
PT2	0
PC1	1
PC2	2
PC3	3
PC4	4
PC5	5
PP1	6
PP2	7
PP3	8

Nota. El punto de partida siempre será identificado por “0”

La ruta obtenida por el Algoritmo del Vecino más Cercano con los índices de la Tabla 10, sería la siguiente manera: 0 - 1 - 2 - 3 - 4 - 7 - 5 - 8 - 6 – 0.

El Paso 0 comienza con seleccionar el recorrido de inicio (o recorrido de la iteración 0) y determinar el periodo de tenencia en la lista tabú:

- Recorrido Inicial: 0 - 1 - 2 - 3 - 4 - 7 - 5 - 8 - 6 – 0
- Periodo de Tenencia: 2 iteraciones.

Los pasos 1, 2 y 3 se van a realizar para cada iteración. En total se harán quince iteraciones para luego seleccionar la mejor solución de todas.

El Paso 1 de la primera iteración comienza con realizar la inversión de subrecorridos del recorrido de la iteración anterior (iteración 0), se debe evitar hacer la inversión con el punto de partida (0-1 o 6-0).

La inversión de los subrecorridos consistirá en invertir tramos de dos puntos adyacentes hasta siete puntos adyacentes, debido a que si se invirtiese los ocho puntos no cambiaría en nada la distancia del recorrido de la iteración anterior, debido a que la matriz de distancia asociada es simétrica. En forma generalizada las inversiones de los subrecorridos serán (n-1), siendo “n” el número total de destinos involucrados en cada ruta. En la Tabla 10.1 se ven las inversiones.

El Paso 2 de la primera iteración consiste en identificar de todas las inversiones aquel recorrido con la menor distancia. De la Tabla 10.1, la inversión de 7-8-5-6 genera el menor recorrido de todos, por lo que es la solución de la primera iteración, adicionalmente se determina los arcos eliminados e añadidos y los elementos de la lista Tabú. Queda lo siguiente al final de esta primera iteración:

- Solución: 0-1-2-3-4-6-8-5-7-0
- Distancia: 2597 metros
- Arcos eliminados: 4-7 y 6-0
- Arcos añadidos: 4-6 y 7-0
- Lista Tabú: 4-6 (permanecerá 2 iteraciones) y 7-0 (permanecerá 2 iteraciones)

Tabla 10.1

Inversiones de subrecorridos de la primera iteración

Ite.	Inversión	Recorrido	Distancia (metros)
0	-	0-1-2-3-4-7-5-8-6 -0	2692
	1-2	0-2-1-3-4-7-5-8-6-0	2972
	2-3	0-1-3-2-4-7-5-8-6-0	2995
	3-4	0-1-2-4-3-7-5-8-6-0	3115
	4-7	0-1-2-3-7-4-5-8-6-0	2787
	7-5	0-1-2-3-4-5-7-8-6-0	2857
	5-8	0-1-2-3-4-7-8-5-6-0	3038
	8-6	0-1-2-3-4-7-5-6-8-0	2688
	1-2-3	0-3-2-1-4-7-5-8-6-0	3283
	2-3-4	0-1-4-3-2-7-5-8-6-0	3406
	3-4-7	0-1-2-7-4-3-5-8-6-0	3270
	4-7-5	0-1-2-3-5-7-4-8-6-0	3058
	7-5-8	0-1-2-3-4-8-5-7-6-0	3274
	5-8-6	0-1-2-3-4-7-6-8-5-0	2698
1	1-2-3-4	0-4-3-2-1-7-5-8-6-0	3674
	2-3-4-7	0-1-7-4-3-2-5-8-6-0	3546
	3-4-7-5	0-1-2-5-7-4-3-8-6-0	3543
	4-7-5-8	0-1-2-3-8-5-7-4-6-0	3415
	7-5-8-6	0-1-2-3-4-6-8-5-7-0	2597
	1-2-3-4-7	0-7-4-3-2-1-5-8-6-0	3842
	2-3-4-7-5	0-1-5-7-4-3-2-8-6-0	3830
	3-4-7-5-8	0-1-2-8-5-7-4-3-6-0	3911
	4-7-5-8-6	0-1-2-3-6-8-5-7-4-0	2692
	1-2-3-4-7-5	0-5-7-4-3-2-1-8-6-0	4109
	2-3-4-7-5-8	0-1-8-5-7-4-3-2-6-0	4192
	3-4-7-5-8-6	0-1-2-6-8-5-7-4-3-0	2692
	1-2-3-4-7-5-8	0-8-5-7-4-3-2-1-6-0	4494
	2-3-4-7-5-8-6	0-1-6-8-5-7-4-3-2-0	2676

El Paso 3 de la primera iteración es verificar si se llegó a una condición de parada (como se indicó será cuando se hayan hecho quince iteraciones) o volver al Paso 1. Se va a retornar al Paso 1.

El Paso 1 de la segunda iteración comienza con realizar la inversión de subrecorridos del recorrido de la iteración anterior (iteración 1). En la Tabla 10.2 se muestra.

Tabla 10.2

Inversiones de subrecorridos de la segunda iteración

Ite.	Inversión	Recorrido	Distancia (metros)
1	-	0-1-2-3-4-6-8-5-7-0	2597
	1-2	0-2-1-3-4-6-8-5-7-0	2877
	2-3	0-1-3-2-4-6-8-5-7-0	2900
	3-4	0-1-2-4-3-6-8-5-7-0	3097
	4-6	0-1-2-3-6-4-8-5-7-0	3334
	6-8	0-1-2-3-4-8-6-5-7-0	2617
	8-5	0-1-2-3-4-6-5-8-7-0	2943
	5-7	0-1-2-3-4-6-8-7-5-0	2846
	1-2-3	0-3-2-1-4-6-8-5-7-0	3188
	2-3-4	0-1-4-3-2-6-8-5-7-0	3397
	3-4-6	0-1-2-6-4-3-8-5-7-0	3820
	4-6-8	0-1-2-3-8-6-4-5-7-0	2957
	6-8-5	0-1-2-3-4-5-8-6-7-0	2614
	8-5-7	0-1-2-3-4-6-7-5-8-0	3250
2	1-2-3-4	0-4-3-2-1-6-8-5-7-0	3686
	2-3-4-6	0-1-6-4-3-2-8-5-7-0	4114
	3-4-6-8	0-1-2-8-6-4-3-5-7-0	3445
	4-6-8-5	0-1-2-3-5-8-6-4-7-0	2761
	6-8-5-7	0-1-2-3-4-7-5-8-6-0	2692
	1-2-3-4-6	0-6-4-3-2-1-8-5-7-0	4388
	2-3-4-6-8	0-1-8-6-4-3-2-5-7-0	3725
	3-4-6-8-5	0-1-2-5-8-6-4-3-7-0	3183
	4-6-8-5-7	0-1-2-3-7-5-8-6-4-0	2615
	1-2-3-4-6-8	0-8-6-4-3-2-1-5-7-0	4020
	2-3-4-6-8-5	0-1-5-8-6-4-3-2-7-0	3465
	3-4-6-8-5-7	0-1-2-7-5-8-6-4-3-0	2606
	1-2-3-4-6-8-5	0-5-8-6-4-3-2-1-7-0	3740
	2-3-4-6-8-5-7	0-1-7-5-8-6-4-3-2-0	2569

El Paso 2 de la segunda iteración consiste en identificar de todas las inversiones el recorrido con la menor distancia. De la Tabla 10.2, la inversión de 2-3-4-6-8-5-7 genera el menor recorrido de todos. Se evalúa lo siguiente en esta segunda iteración:

- Solución: 0-1-7-5-8-6-4-3-2-0
- Distancia: 2569 metros
- Arcos eliminados: 1-2 y 7-0

- Arcos añadidos: 1-7 y 2-0

Como los arcos añadidos del recorrido seleccionado no coinciden con los arcos que están presentes en la lista tabú (4-6 y 7-0), esta es considerada como la solución de la segunda iteración. En el caso que los arcos eliminados hayan coincidido con los elementos de la lista Tabú, este se le llamaría *solución Tabú* y se debe buscar otro recorrido con menor distancia del conjunto de inversiones de la segunda iteración (ver Tabla10.2). Si en caso la solución Tabú tenga la distancia de recorrido menor a la de la iteración anterior, se puede omitir que sea solución Tabú y considerar como la solución de la segunda iteración. Finalmente, en la situación donde no se cumple ninguno de los escenarios mencionados, el algoritmo se termina y de las iteraciones hechas se selecciona la mejor solución.

Se establece lo siguiente al final de la segunda iteración:

- Solución: 0-1-7-5-8-6-4-3-2-0
- Distancia: 2569 metros
- Arcos eliminados: 1-2 y 7-0
- Arcos añadidos: 1-7 y 2-0
- Lista Tabú: 1-7 (permanecerá 2 iteraciones), 2-0 (permanecerá 2 iteraciones), 4-6 (permanecerá 1 iteración) y 7-0 (permanecerá 1 iteración).

Los resultados de todas las iteraciones se encuentran en la Tabla 10.3.

Tabla 10.3

Resultados de todas las iteraciones

Ite.	Recorrido	Distancia (metros)
0	0-1-2-3-4-7-5-8-6-0	2692
1	0-1-2-3-4-6-8-5-7-0	2597
2	0-1-7-5-8-6-4-3-2-0	2569
3	0-1-7-6-8-5-4-3-2-0	2586
4	0-1-7-8-6-5-4-3-2-0	2552
5	0-1-3-4-5-6-8-7-2-0	2578
6	0-1-7-8-6-5-4-3-2-0	2552
7	0-1-2-3-4-5-6-8-7-0	2580
8	0-1-7-8-6-5-4-3-2-0	2552
9	0-1-3-4-5-6-8-7-2-0	2578
10	0-1-7-8-6-5-4-3-2-0	2552
11	0-1-2-3-4-5-6-8-7-0	2580
12	0-1-7-8-6-5-4-3-2-0	2552
13	0-1-3-4-5-6-8-7-2-0	2578
14	0-1-7-8-6-5-4-3-2-0	2552
15	0-1-2-3-4-5-6-8-7-0	2580

Según los resultados mostrados en la Tabla 10.3, la mejor solución se encuentra en las iteraciones 4,6,10,12 y 14. Se va a elegir cualquiera y reemplazando los índices numéricos por sus valores originales, el resultado final del algoritmo es:

- Ruta: PT2 – PC1 – PP2 – PP3 – PP1 – PC5 – PC4 – PC3 – PC2 – PT2
- Distancia: 2552 metros

En el caso que el algoritmo del Vecino más Cercano haya brindado una segunda solución o más de ellas, de la misma forma se le aplica el procedimiento del Algoritmo de la Búsqueda Tabú a cada uno.

Diseño de rutas con el algoritmo Genético

El Algritmo Genético (AG) no necesita de soluciones iniciales brindadas por heurísticas, porque puede construirlas por sí mismo. En esta investigación, estas soluciones se hacen mediante procedimientos aleatorios.

Antes de aplicar el algoritmo genético, se va a determinar de manera aleatoria los destinos que estarán involucrados en la construcción de cada ruta. Por ejemplo, si la restricción es diseñar rutas que involucren visitar como máximo ocho puntos de un total

de doce que hayan sido seleccionados, lo que se hará es seleccionar aleatoriamente los ocho primeros destinos para la primera ruta y luego los cuatro faltantes para la segunda, pero se debe de tener en cuenta que cada destino debe estar presente en una sola ruta. De la misma forma, si la restricción es visitar como máximo cuatro puntos de un total de doce que hayan sido seleccionados, se comienza por seleccionar aleatoriamente cuatros destinos para la primera ruta y así sucesivamente.

Una vez determinados los destinos por ruta, se tomará en cuenta el orden en que hayan ido apareciendo secuencialmente con el fin de generar una trayectoria inicial, esto no afectará a los resultados finales porque se le va a aplicar el algoritmo genético a cada una de las rutas construidas aleatoriamente, pero se les aplica el algoritmo a aquellas que contengan más de dos destinos.

En esta investigación se va a realizar el mismo procedimiento hecho por Taha (2012) en el ejemplo que desarrollo en su libro (Ejemplo 11.5-3).

Para el ejemplo que se ha venido desarrollando, todos los puntos deben ser visitados en una sola trayectoria (ocho destinos), por lo que una trayectoria inicial construida aleatoriamente podría ser:

- PT2 - PP1 - PC5 - PC3 - PC4 - PP3 - PC2 - PC1 - PP2 - PT2
- Distancia: 4982 metros

Para cada ruta se va asigna un código genético de tipo numérico. Ver la Tabla 11.

Tabla 11

Índices de los destinos para el Algoritmo Génetico

Puntos	Índice
PT2	0
PP1	1
PC5	2
PC3	3
PC4	4
PP3	5
PC2	6
PC1	7
PP2	8

Nota. El punto de partida siempre será identificado por “0”

La ruta con los índices de la Tabla 11, seria de la siguiente manera: 0-1-2-3-4-5-6-7-8–0. Este será la solución de la iteración cero.

El Paso 0 es generar una población aleatoria, luego evaluar la aptitud de cada uno de sus individuos. La población será un conjunto de seis rutas que se construye aleatoriamente con los puntos de la Tabla 11. Una población seria lo siguiente:

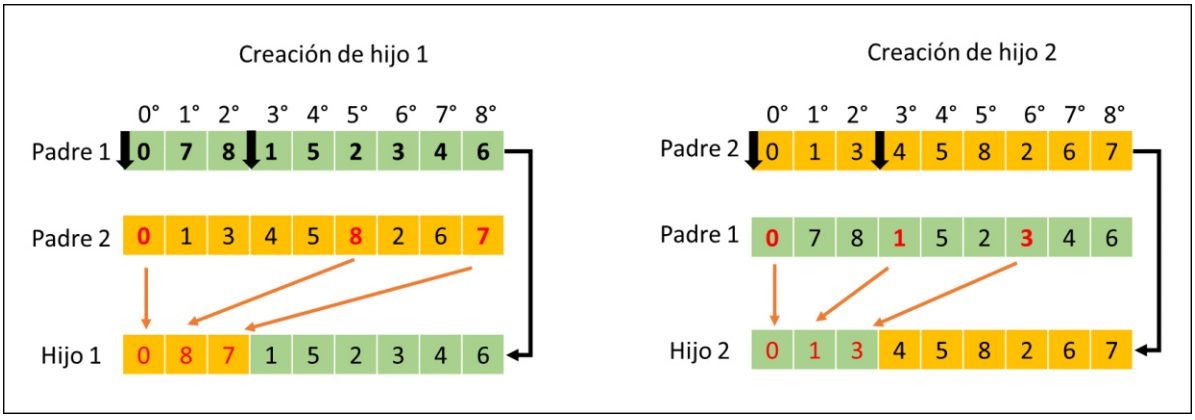
- Cromosoma 1: 0-5-1-6-7-4-8-3-2-0 Distancia: 4709 metros
- Cromosoma 2: 0-2-1-4-3-7-8-6-5-0 Distancia: 5267 metros
- Cromosoma 3: 0-7-8-1-5-2-3-4-6-0 Distancia: 3078 metros
- Cromosoma 4: 0-1-3-4-5-8-2-6-7-0 Distancia: 4085 metros
- Cromosoma 5: 0-8-6-3-4-5-7-2-1-0 Distancia: 5276 metros
- Cromosoma 6: 0-5-8-1-6-7-4-3-2-0 Distancia: 5159 metros

Se va a presentar los pasos para la primera iteración. El Paso 1 consiste en seleccionar dos padres, para el caso en curso dos rutas (o cromosomas) con la mínima distancia de recorrido de todo el conjunto de población, siendo los cromosomas 3 y 4.

El siguiente paso es realizar el cruce de genes de los padres seleccionados para crear dos hijos. Como se mencionó en las bases teóricas se va a utilizar el método del operador de cruce por dos puntos. Los puntos de cruce se seleccionan aleatoriamente, para el ejemplo en curso son los genes que estén ubicados entre la posición 0 y 2 del cromosoma de los padres. En la Figura 10 se muestra lo expuesto.

Figura 10

Creación de hijos para el A. Genético



Nota. Cuando se realiza el cruce de genes, no se toma en cuenta el punto final de la ruta (gen “0”).

En la Figura 10 se ilustra un resumen del procedimiento detallado en las bases teóricas. El proceso de mutación en el gen de los hijos se hace presente cuando se le asocia a cada hijo una probabilidad de variación en los genes (denotado por R). Si la probabilidad es menor a 0.1, entonces sí existe una mutación en sus genes. Las probabilidades de mutación para cada hijo se determinan aleatoriamente. Para el ejemplo se presenta lo siguiente:

- Hijo 1, R = 0.8206
- Hijo 2, R = 0.4205

En este escenario no se presenta mutación, en caso si haya existido se realiza el procedimiento expuesto en las bases teóricas. El procedimiento final del Paso 1 es identificar de la población los dos cromosomas menos aptos que serian las dos rutas con la distancia más alta y estas reemplazarlas por los hijos. Para el ejemplo se van a reemplazar los cromosomas 2 y 5 por los hijos encontrados. Los resultados se muestran en la Tabla 11.1.

Tabla 11.1

Resultados de la primera iteración

Ite.	Población		Padres	Cruce	Hijos	Probabilidad Mutación	Nueva Población	
	Trayectoria	Dist.					Trayectoria	Dist.
0	0-1-2-3-4-5-6-7-8–0	4982	----	----	----	----	----	----
1	0-5-1-6-7-4-8-3-2-0	4709	Padre 1:	0 y 2	Hijo 1:		0-5-1-6-7-4-8-3-2-0	4709
	0-2-1-4-3-7-8-6-5-0	5267	0-7-8-1-5-2-3-4-6-0		0-8-7-1-5-2-3-4-6	R = 0,4205	0-8-7-1-5-2-3-4-6-0	4227
	0-7-8-1-5-2-3-4-6-0	3078				(No hay)	0-7-8-1-5-2-3-4-6-0	3078
	0-1-3-4-5-8-2-6-7-0	4085	Padre 2:		Hijo2:		0-1-3-4-5-8-2-6-7-0	4085
	0-8-6-3-4-5-7-2-1-0	5276	0-1-3-4-5-8-2-6-7-0		0-1-3-4-5-8-2-6-7	R = 0,8206	0-1-3-4-5-8-2-6-7-0	4085
	0-5-8-1-6-7-4-3-2-0	5159				(No hay)	0-5-8-1-6-7-4-3-2-0	5159

Nota. En caso exista mutación, se cambian los genes según las posiciones indicadas. Dist. (distancia) en metros.

De la Tabla 11.1, la solución es la ruta: 0-7-8-1-5-2-3-4-6-0, con una distancia de 3078 metros. El Paso 2 es verificar la condición de parada, en este ejemplo son cuatro iteraciones. En la Tabla 11.2 se ven los resultados restantes.

Tabla 11.2

Resultados de las iteraciones faltantes

Ite.	Población		Padres	Cruce	Hijos	Probabilidad Mutación	Nueva Población	
	Trayectoria	Dist.					Trayectoria	Dist.
0	0-1-2-3-4-5-6-7-8–0	4982	----	----	----	----	----	----
2	0-5-1-6-7-4-8-3-2-0	4709	Padre 1:	1 y 2	Hijo 1:	R = 0,0019 (1° y 3°)	0-1-7-8-5-2-3-4-6-0	4850
	0-8-7-1-5-2-3-4-6-0	4227	0-7-8-1-5-2-3-4-6-0		0-8-7-1-5-2-3-4-6		0-8-7-1-5-2-3-4-6-0	4227
	0-7-8-1-5-2-3-4-6-0	3078			0-1-7-8-5-2-3-4-6		0-7-8-1-5-2-3-4-6-0	3078
	0-1-3-4-5-8-2-6-7-0	4085	Padre 2:		Hijo2:		0-1-3-4-5-8-2-6-7-0	4085
	0-1-3-4-5-8-2-6-7-0	4085	0-1-3-4-5-8-2-6-7-0		0-1-3-4-5-8-2-6-7	R = 0,8074 (No hay)	0-1-3-4-5-8-2-6-7-0	4085
	0-5-8-1-6-7-4-3-2-0	5159					0-1-3-4-5-8-2-6-7-0	4085
3	0-1-7-8-5-2-3-4-6-0	4850	Padre 1:	4 y 7	Hijo 1:	R = 0,9486 (No hay)	0-1-3-4-8-5-2-6-7-0	3883
	0-8-7-1-5-2-3-4-6-0	4227	0-7-8-1-5-2-3-4-6-0		0-7-8-1-3-4-5-2-6		0-7-8-1-3-4-5-2-6-0	3814
	0-7-8-1-5-2-3-4-6-0	3078					0-7-8-1-5-2-3-4-6-0	3078
	0-1-3-4-5-8-2-6-7-0	4085	Padre 2:		Hijo2:		0-1-3-4-5-8-2-6-7-0	4085
	0-1-3-4-5-8-2-6-7-0	4085	0-1-3-4-5-8-2-6-7-0		0-1-3-4-5-8-2-6-7	R = 0,8328 (No hay)	0-1-3-4-5-8-2-6-7-0	4085
	0-1-3-4-5-8-2-6-7-0	4085					0-1-3-4-5-8-2-6-7-0	4085
4	0-1-3-4-8-5-2-6-7-0	3883	Padre 1:	3 y 4	Hijo 1:	R = 0,4855 (No hay)	0-1-3-4-8-5-2-6-7-0	3883
	0-7-8-1-3-4-5-2-6-0	3814	0-7-8-1-5-2-3-4-6-0		0-7-8-1-5-2-3-4-6		0-7-8-1-3-4-5-2-6-0	3814
	0-7-8-1-5-2-3-4-6-0	3078					0-7-8-1-5-2-3-4-6-0	3078
	0-1-3-4-5-8-2-6-7-0	4085	Padre 2:		Hijo2:		0-7-8-6-3-4-5-2-1-0	3852
	0-1-3-4-5-8-2-6-7-0	4085	0-1-3-4-5-8-2-6-7-0		0-7-8-1-3-4-5-2-6	R = 0,0267 (3° y 8°)	0-7-8-1-5-2-3-4-6-0	3078
	0-1-3-4-5-8-2-6-7-0	4085			0-7-8-6-3-4-5-2-1		0-1-3-4-5-8-2-6-7-0	4085

Nota. En caso exista mutación, se cambian los genes según las posiciones indicadas. Dist. (distancia) en metros.

De la Tabla 11.2 se muestra que los resultados empiezan a ser repetitivos, es probable que sea necesario realizar más iteraciones para encontrar una mejor solución. Sin embargo, se va a hacer realizar una segunda corrida (o segundo intento), con otros números aleatorios para la generación de la población inicial, cruce de genes y probabilidad de mutación.

Tabla 11.3

Resultados de las iteraciones de la segunda corrida

Ite.	Población		Padres	Cruce	Hijos	Probabilidad Mutación	Nueva Población	
	Trayectoria	Dist.					Trayectoria	Dist.
0	0-1-2-3-4-5-6-7-8–0	4982	----	----	----	----	----	----
1	0-7-1-4-3-6-2-8-5-0	4359	Padre 1:	2 y 7	Hijo 1:		0-7-1-4-3-6-2-8-5-0	4359
	0-6-4-3-1-8-7-2-5-0	4872	0-7-1-4-3-6-2-8-5		0-7-3-4-2-8-6-1-5	R = 0,0958	0-6-4-3-1-8-7-2-5-0	4872
	0-4-2-8-6-5-7-3-1-0	5556			0-7-3-4-5-8-6-1-2	(4° y 8°)	0-7-3-4-5-8-6-1-2-0	4118
	0-3-4-2-8-5-6-1-7-0	4374	Padre 2:		Hijo2:		0-3-4-2-8-5-6-1-7-0	4374
	0-4-8-7-2-3-5-1-6-0	4666	0-3-4-2-8-5-6-1-7		0-3-1-4-6-2-8-5-7	R = 0,7139	0-4-8-7-2-3-5-1-6-0	4666
	0-1-4-2-7-5-3-6-8-0	5616				(No hay)	0-3-1-4-6-2-8-5-7-0	4348
2	0-7-1-4-3-6-2-8-5-0	4359	Padre 1:	6 y 8	Hijo 1:		0-7-1-4-3-6-2-8-5-0	4359
	0-6-4-3-1-8-7-2-5-0	4872	0-7-3-4-5-8-6-1-2		0-7-3-4-5-8-1-6-2	R = 0,8672	0-3-1-4-8-2-5-6-7-0	3411
	0-7-3-4-5-8-6-1-2-0	4118				(No hay)	0-7-3-4-5-8-6-1-2-0	4118
	0-3-4-2-8-5-6-1-7-0	4374	Padre 2:		Hijo2:		0-3-4-2-8-5-6-1-7-0	4374
	0-4-8-7-2-3-5-1-6-0	4666	0-3-1-4-6-2-8-5-7		0-3-1-4-6-2-5-8-7	R = 0,0686	0-7-3-4-5-8-1-6-2-0	4394
	0-3-1-4-6-2-8-5-7-0	4348			0-3-1-4-5-2-6-8-7	(4° y 6°)	0-3-1-4-6-2-8-5-7-0	4348
3	0-7-1-4-3-6-2-8-5-0	4359	Padre 1:	3 y 6	Hijo 1:		0-7-1-4-3-6-2-8-5-0	4359
	0-3-1-4-8-2-5-6-7-0	3411	0-3-1-4-8-2-5-6-7		0-3-1-4-5-8-2-6-7	R = 0,99	0-3-1-4-8-2-5-6-7-0	3411
	0-7-3-4-5-8-6-1-2-0	4118				(No hay)	0-7-3-4-5-8-6-1-2-0	4118
	0-3-4-2-8-5-6-1-7-0	4374	Padre 2:		Hijo2:		0-3-1-4-5-8-2-6-7-0	3585
	0-7-3-4-5-8-1-6-2-0	4394	0-3-4-2-8-5-6-1-7-0		0-7-3-4-8-5-6-1-2	R = 0,2253	0-7-3-4-8-5-6-1-2-0	4186
	0-3-1-4-6-2-8-5-7-0	4348				(No hay)	0-3-1-4-6-2-8-5-7-0	4348
4	0-7-1-4-3-6-2-8-5-0	4359	Padre 1:	0 y 4	Hijo 1:		0-3-1-4-5-8-2-6-7-0	3585
	0-3-1-4-8-2-5-6-7-0	3411	0-3-1-4-8-2-5-6-7		0-3-1-4-8-2-5-6-7	R = 0,8138	0-3-1-4-8-2-5-6-7-0	3411
	0-7-3-4-5-8-6-1-2-0	4118				(No hay)	0-7-3-4-5-8-6-1-2-0	4118
	0-3-1-4-5-8-2-6-7-0	3585	Padre 2:		Hijo2:		0-3-1-4-5-8-2-6-7-0	3585
	0-7-3-4-8-5-6-1-2-0	4186	0-3-1-4-5-8-2-6-7		0-3-1-4-5-8-2-6-7	R = 0,7607	0-7-3-4-8-5-6-1-2-0	4186
	0-3-1-4-6-2-8-5-7-0	4348				(No hay)	0-3-1-4-8-2-5-6-7-0	3411

Nota. En caso exista mutación, se cambian los genes según las posiciones indicadas. Dist. (distancia) en metros.

Los resultados mostrados en la Tabla 11.3 muestran que las soluciones obtenidos en la segunda corrida no son mejores a los que se obtuvo en un primer intento, sin embargo, se va intentar una tercera y cuarta corrida con la intención de analizarlos. La Tabla 11.4 muestra los resultados de la tercera corrida y la Tabla 11.5 de la cuarta corrida.

Tabla 11.4

Resultados de las iteraciones de la tercera corrida

Ite.	Población		Padres	Cruce	Hijos	Probabilidad Mutación	Nueva Población	
	Trayectoria	Dist.					Trayectoria	Dist.
0	0-1-2-3-4-5-6-7-8-0	4982	----	----	----	----	----	----
1	0-5-4-2-1-6-7-3-8-0	4519	Padre 1:	1 y 2	Hijo 1:		0-5-4-2-1-6-7-3-8-0	4519
	0-3-4-7-1-2-8-5-6-0	4297	0-8-3-6-1-2-5-4-7-0		0-3-8-6-1-2-5-4-7	R = 0,9204	0-3-4-7-1-2-8-5-6-0	4297
	0-8-3-6-1-2-5-4-7-0	3875				(No hay)	0-8-3-6-1-2-5-4-7-0	3875
	0-3-1-7-4-8-5-2-6-0	4458	Padre 2:		Hijo2:		0-3-1-7-4-8-5-2-6-0	4458
	0-4-3-7-1-2-8-5-6-0	4300	0-4-3-7-1-2-8-5-6-0		0-3-1-4-7-1-2-8-5-6	R = 0,1743	0-4-3-7-1-2-8-5-6-0	4300
	0-7-1-4-6-2-8-3-5-0	4977				(No hay)	0-3-8-6-1-2-5-4-7-0	3884
2	0-5-4-2-1-6-7-3-8-0	4519	Padre 1:	2 y 6	Hijo 1:		0-5-4-2-1-6-7-3-8-0	3884
	0-3-4-7-1-2-8-5-6-0	4297	0-8-3-6-1-2-5-4-7-0		0-8-3-6-1-2-5-4-7	R = 0,5979	0-3-4-7-1-2-8-5-6-0	4297
	0-8-3-6-1-2-5-4-7-0	3875				(No hay)	0-8-3-6-1-2-5-4-7-0	3875
	0-3-1-7-4-8-5-2-6-0	4458	Padre 2:		Hijo2:		0-8-3-6-1-2-5-4-7-0	3875
	0-4-3-7-1-2-8-5-6-0	4300	0-3-8-6-1-2-5-4-7-0		0-3-8-6-1-2-5-4-7	R = 0,7517	0-4-3-7-1-2-8-5-6-0	4300
	0-3-8-6-1-2-5-4-7-0	3884				(No hay)	0-3-8-6-1-2-5-4-7-0	3884
3	0-5-4-2-1-6-7-3-8-0	3884	Padre 1:	5 y 7	Hijo 1:		0-5-4-2-1-6-7-3-8-0	3884
	0-3-4-7-1-2-8-5-6-0	4297	0-8-3-6-1-2-5-4-7-0		0-8-3-6-1-2-5-4-7	R = 0,9886	0-3-4-7-1-2-8-5-6-0	4297
	0-8-3-6-1-2-5-4-7-0	3875				(No hay)	0-8-3-6-1-2-5-4-7-0	3875
	0-8-3-6-1-2-5-4-7-0	3875	Padre 2:		Hijo2:		0-8-3-6-1-2-5-4-7-0	3875
	0-4-3-7-1-2-8-5-6-0	4300	0-8-3-6-1-2-5-4-7-0		0-8-3-6-1-2-5-4-7	R = 0,9364	0-4-3-7-1-2-8-5-6-0	4300
	0-3-8-6-1-2-5-4-7-0	3884				(No hay)	0-3-8-6-1-2-5-4-7-0	3884

Nota. En caso exista mutación, se cambian los genes según las posiciones indicadas. Dist. (distancia) en metros.

Tabla 11.5

Resultados de las iteraciones de la cuarta corrida

Ite.	Población		Padres	Cruce	Hijos	Probabilidad Mutación	Nueva Población	
	Trayectoria	Dist.					Trayectoria	Dist.
0	0-1-2-3-4-5-6-7-8–0	4982	----	----	----	----	----	----
1	0-5-1-6-7-2-3-8-4-0	4702	Padre 1:	0 y 4	Hijo 1:		0-5-1-6-7-2-3-8-4-0	4702
	0-1-4-2-7-5-3-6-8-0	5616	0-8-7-3-6-2-1-5-4		0-6-7-3-8-2-1-5-4	R = 0,932	0-6-7-3-8-2-1-5-4-0	2915
	0-6-7-2-3-1-8-4-5-0	4547				(No hay)	0-6-7-2-3-1-8-4-5-0	4547
	0-4-2-8-6-5-7-3-1-0	5556	Padre 2:		Hijo2:		0-7-3-6-2-1-8-4-5-0	3736
	0-5-8-6-3-7-2-1-4-0	4678	0-6-7-2-3-1-8-4-5		0-7-3-6-2-1-8-4-5	R = 0,76	0-5-8-6-3-7-2-1-4-0	4678
	0-8-7-3-6-2-1-5-4-0	4065				(No hay)	0-8-7-3-6-2-1-5-4-0	4065
2	0-5-1-6-7-2-3-8-4-0	4702	Padre 1:	1 y 6	Hijo 1:		0-6-7-3-8-2-1-4-5-0	3628
	0-6-7-3-8-2-1-5-4-0	2915	0-6-7-3-8-2-1-5-4		0-7-3-6-2-1-8-5-4	R = 0,841	0-6-7-3-8-2-1-5-4-0	2915
	0-6-7-2-3-1-8-4-5-0	4547				(No hay)	0-6-7-2-3-1-8-4-5-0	4547
	0-7-3-6-2-1-8-4-5-0	3736	Padre 2:		Hijo2:		0-7-3-6-2-1-8-4-5-0	3736
	0-5-8-6-3-7-2-1-4-0	4678	0-7-3-6-2-1-8-4-5		0-6-7-3-8-2-1-4-5	R = 0,8018	0-7-3-6-2-1-8-5-4-0	3551
	0-8-7-3-6-2-1-5-4-0	4065				(No hay)	0-8-7-3-6-2-1-5-4-0	4065

Nota. En caso exista mutación, se cambian los genes según las posiciones indicadas. Dist. (distancia) en metros.

La Tabla 11.5 muestra que los resultados de la cuarta corrida son mejores de todos los intentos.

Tabla 11.6

Solución final de cada corrida

# Corrida	# Iteraciones hechas	Solución final	
		Ruta	Distancia
1	4	0-7-8-1-5-2-3-4-6-0	3078 metros
2	4	0-3-1-4-8-2-5-6-7-0	3411 metros
3	3	0-8-3-6-1-2-5-4-7-0	3875 metros
4	2	0-6-7-3-8-2-1-5-4-0	2915 metros

Según los resultados mostrados en la Tabla 11.6, la mejor solución se encuentra en la cuarta corrida y reemplazando el el código genético por sus valores originales, el resultado final del algoritmo es:

- Ruta: PT2 – PC2 – PC1 – PC3 – PP2 – PC5 – PP1 – PP3 – PC4 – PT2
- Distancia: 2915 metros

La solución que se ha encontrado no se puede asegurar que sea la mejor porque como se vio durante el desarrollo, dependiendo de los valores aleatorios que vayan apareciendo en las iteraciones, los resultados van a ser distintos en cada corrida o intentos.

Para realizar una búsqueda más extensa en este algoritmo, mediante el recurso del lenguaje de programación, se va a realizar diez corridas donde se realicen cuatrocientas iteraciones en cada una de ellas, con el fin de seleccionar la mejor solución de todos los intentos. Si bien es cierto realizar el procedimiento mencionado tampoco asegura encontrar el mejor resultado del algoritmo, por lo menos se hará una exploración más profunda con el objetivo de encontrar una solución cercano al óptimo.

5.3. Aplicativo desktop especializado en diseñar rutas de patrullaje vehicular

En esta sección se va a detallar el proceso de funcionamiento del aplicativo desktop elaborado para esta investigación. El aplicativo fue hecho con el lenguaje de programación Visual Basic para WindowsForm dentro de Visual Studio 2019. Se instaló un paquete de Nuget llamado GMap.NET.WindowsForms del autor radioman para la visualización de mapas de Google.

Para una mejor explicación, se va a presentar el procedimiento hecho para construir rutas a la CPNP Manzano. El formulario inicial del aplicativo se muestra en la Figura 11.

Figura 11

Ventana de Pantalla de Inicio del aplicativo



Nota. Los puntos negros indican: en (1) se encuentra la lista de los nombres de las comisarias del distrito del Rímac, en (2) un botón que al ejecutarse da inicio al siguiente formulario y en (3) se ubica otro botón que al ejecutarse cierra el apicativo completo.

Los pasos a seguir por el usuario es el siguiente, de la Figura 11 se debe elegir una comisaria como se muestra en la Figura 11.1, luego se ejecuta el botón Iniciar.

Figura 11.1

Selección de la comisaria para generar rutas en el aplicativo



Nota. Los códigos de programación se encuentra en Anexos 10.1.

De la Figura 11.1, como se indicó al inicio de esta sección, en busca de una mejor explicación se va a seleccionar la comisaria de Manzano, esto se ve en la Figura 11.2.

Figura 11.2

Selección de una comisaria para generar rutas en el aplicativo

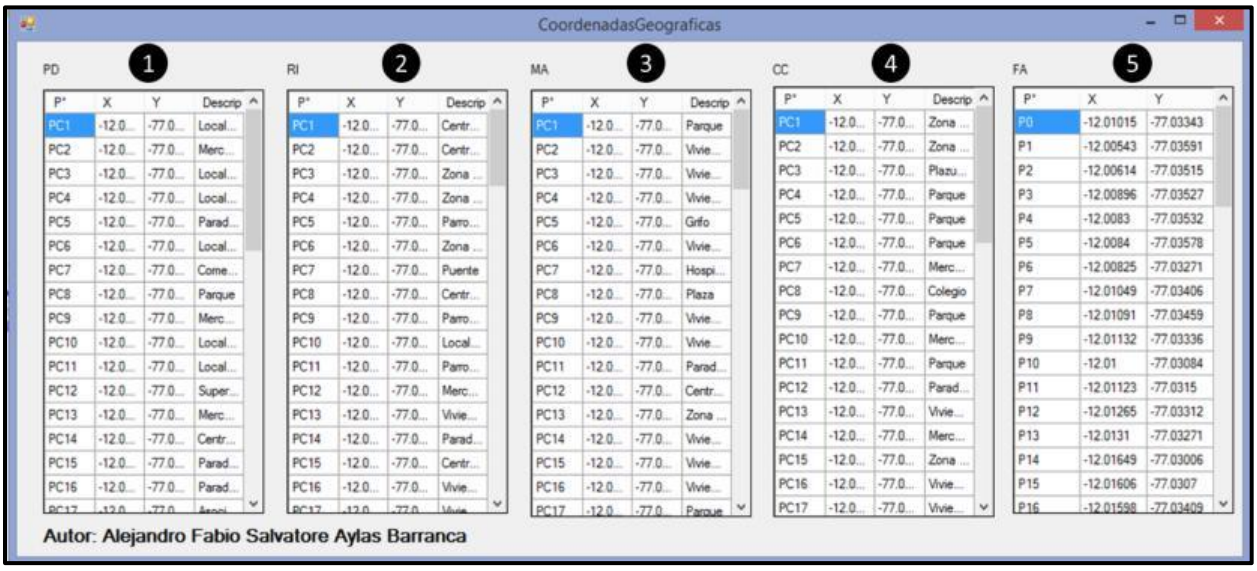


En la Figura 11.2, cuando el botón “Iniciar” es ejecutado, se operan dos formularios, el primero es llamado *CoordenadasGeograficas* y el segundo *Comisaria*. Sin embargo, solo es visible para el usuario el segundo formulario mencionado.

El formulario *CoordenadasGeograficas,* como se ve en la Figura 11.3, tiene como finalidad mantener el registro de las coordenadas geográficas de todos los lugares que son considerados a visitar durante el patrullaje vehicular (puntos críticos y puntos de protección) durante el tiempo de uso del aplicativo. El registro completo se encuentra en Anexo 9.

Figura 11.3

Ventana del formulario CoordenadasGeograficas



Nota. En la imagen muestra en las coordenadas geográficas de los puntos, en (1) lo referido a la CPNP Piedra Liza, en (2) lo referido a CPNP Rímac, en (3) lo referido a CPNP Manzano, en (4) lo referido a CPNP Ciudad y Campo, en (5) lo referido a CPNP Flor de Amancaes. Los códigos de programación asociados a este formulario están en Anexos 10.2.

El formulario *Comisaria* es el único en aparecer, el objetivo es presentar el área donde se va a realizar el patrullaje y que el usuario determine el sector (es) y los puntos a patrullar. En la Figura 11.4 se muestra el formulario para la elección de la CPNP Manzano.

Figura 11.4

Ventana del formulario comisaria



Nota. Los puntos negros indican los controles disponibles para el usuario, en (1) se muestra el nombre de la comisaría que fue seleccionado en el paso anterior, en (2) está el mapa del área de jurisdicción divido por sectores, en (3) se ubica una sección para que el usuario elija el sector o sectores destinados a patrullar por un vehículo, en (4) se presenta una leyenda de los puntos (crítico o protección) que van a aparecer en el mapa, en (5) se va a mostrar una tabla de los puntos críticos y puntos protección establecidos en el sector (es) seleccionado (s) para que el usuario seleccione los destinos a visitar, en (6) se presenta una leyenda de los puntos de inicio (puesto táctico) que van a aparecer en el mapa, en (7) se va a mostrar una tabla de los puestos tácticos establecidos en el sector (es) seleccionado (s) para que el usuario seleccione un punto de partida, en (8) está una sección para indicar el número máximo de puntos a visitar por trayectoria y en (9) está ubicado un botón que al accionarse va a comenzar a ejecutar los procedimientos de los algoritmos matemáticos. Los códigos de programación asociados a este formulario están en Anexos 10.3.

Los pasos a seguir por el usuario son los siguiente, de la Figura 11.4 en (3) se tiene que seleccionar el sector o sectores que serán destinados a patrullar por un

vehiculo, luego se ejecuta el botón aledaño para que en (2) se visualice el mapa con los puntos críticos y protección establecidos en el sector o sectores seleccionados, así mismo en (5) se va a visualizar una tabla de tres columnas, en la primera columna para cada fila se muestra los índices de los puntos, en la segunda columna para cada fila o punto, está un control de selección o checkbox y en la tercera columna el nombre de la localidad cercana al punto. El objetivo de una columna de checkbox es que el usuario seleccione quienes son los puntos a visitar durante el patrullaje. Luego en (7) se presenta una tabla similar a lo mencionado para los puestos táctico para seleccionar que punto será el inicio del recorrido, en (8) se debe indicar el número máximo de puntos críticos por trayectoria, se va a considerar que el mínimo sea cuatro destinos. Se ejecuta en (9) el botón y este empezará a construir las rutas con los algoritmos. La Figura 11.5 se muestra el formulario *Comisaria* con la información para diseñar rutas a la CPNP Manzano.

Figura 11.5

Ventana del formulario comisaria para CPNP Manzano



Nota. En el mapa se seleccionó todos los puntos disponibles y visitar como máximo seis destinos por ruta.

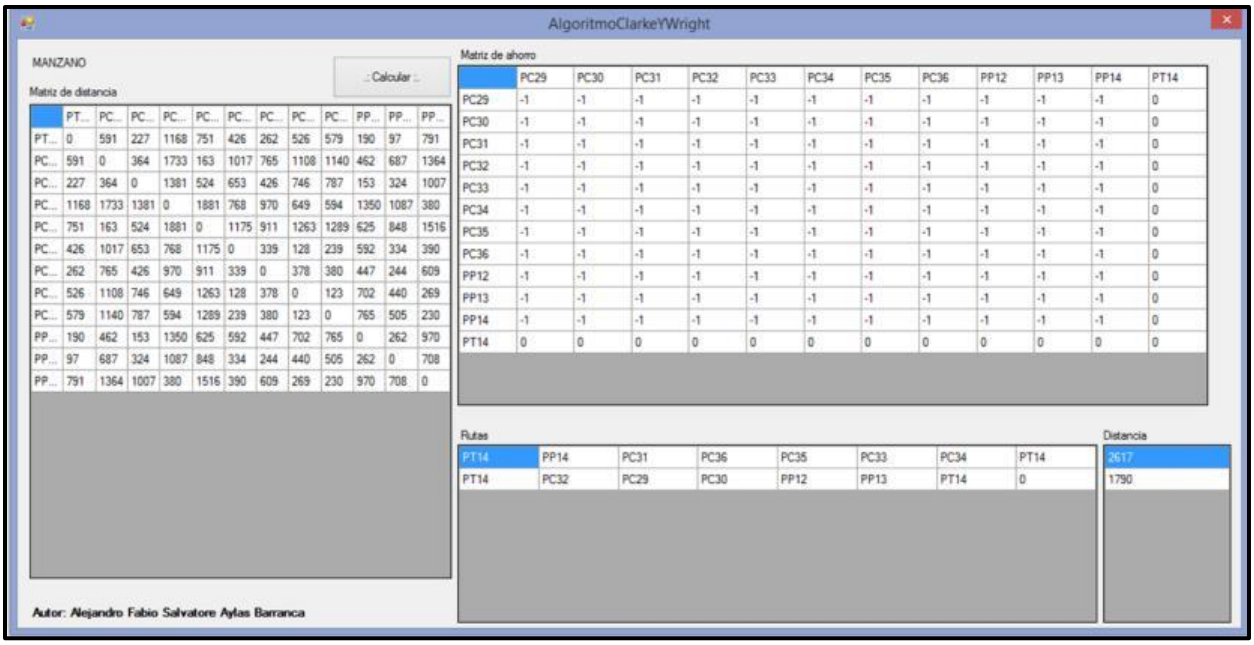
De la Figura 11.5, una vez completada la información, se ejecuta el botón “Rutas”, seguido el aplicativo identifica los puntos seleccionados (destinos y punto de partida) que

van a ser considerados para el diseño de rutas, así mismo almacena el número máximo de puntos a insertar en una ruta.

Con la información recopilada, se crea la matriz de distancias y esta junto con el valor del número máximo de puntos por ruta, son mandados a cada formulario correspondiente a los algoritmos, luego los formularios: *AlgoritmoClarkeYWright*, *AlgoritmoVecinoCercano*, *AlgoritmoBusquedaTabu* y *AlgoritmoGenetico* comienzan sus procedimientos en la construcción de las rutas. Las operaciones de estos formularios no son visibles para el usuario.

Figura 11.6

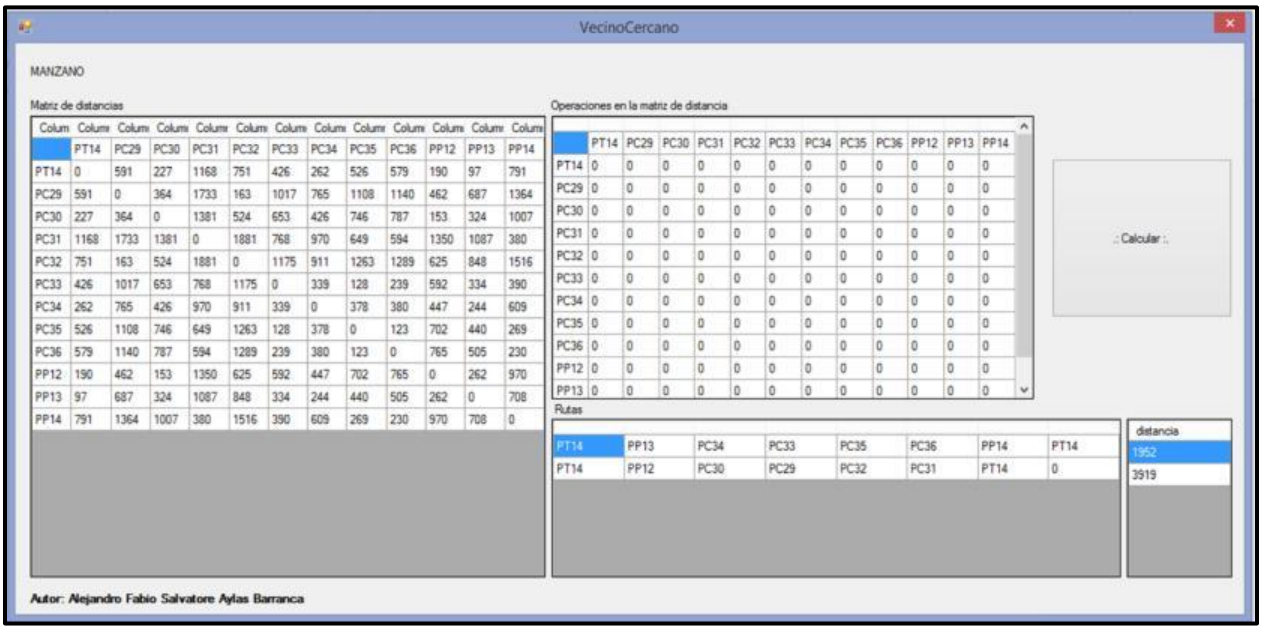
Ventana del formulario AlgoritmoClarkeYWright



Nota. Se ve las operaciones del algoritmo. Los códigos de programación se encuentran en el Anexo 10.4.

Figura 11.7

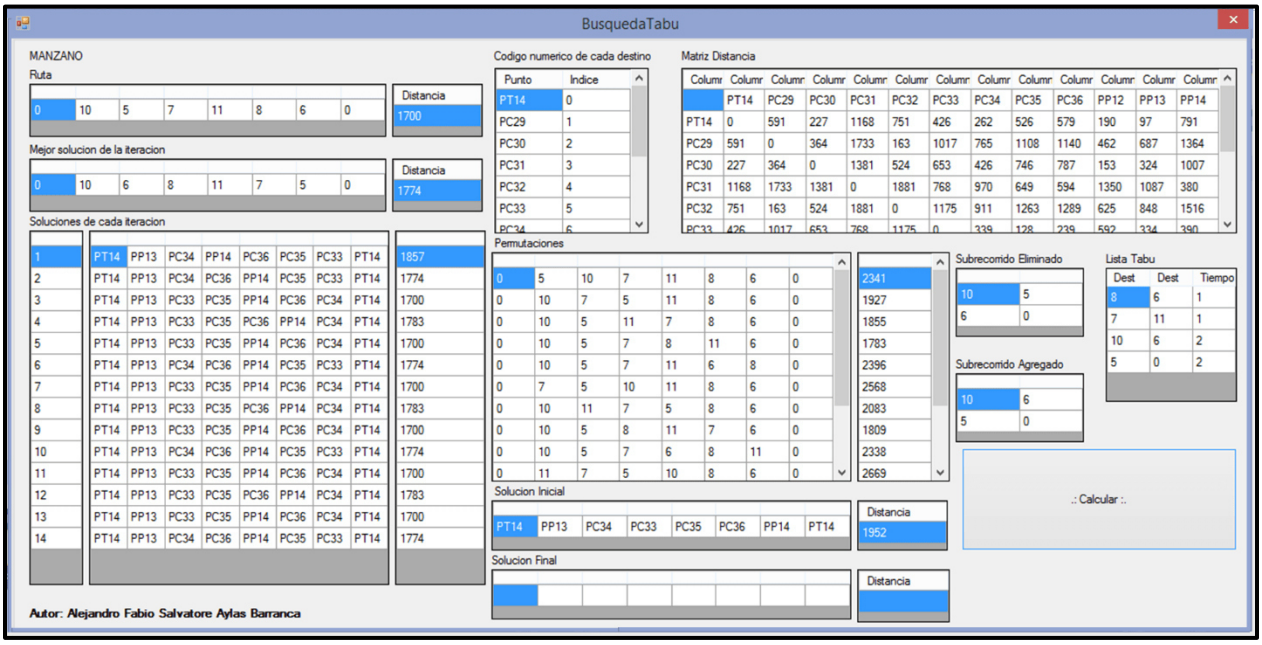
Ventana del formulario AlgoritmoVecinoCercano



Nota. Se ve las operaciones del algoritmo. Los códigos de programación se encuentran en el Anexo 10.5.

Figura 11.8

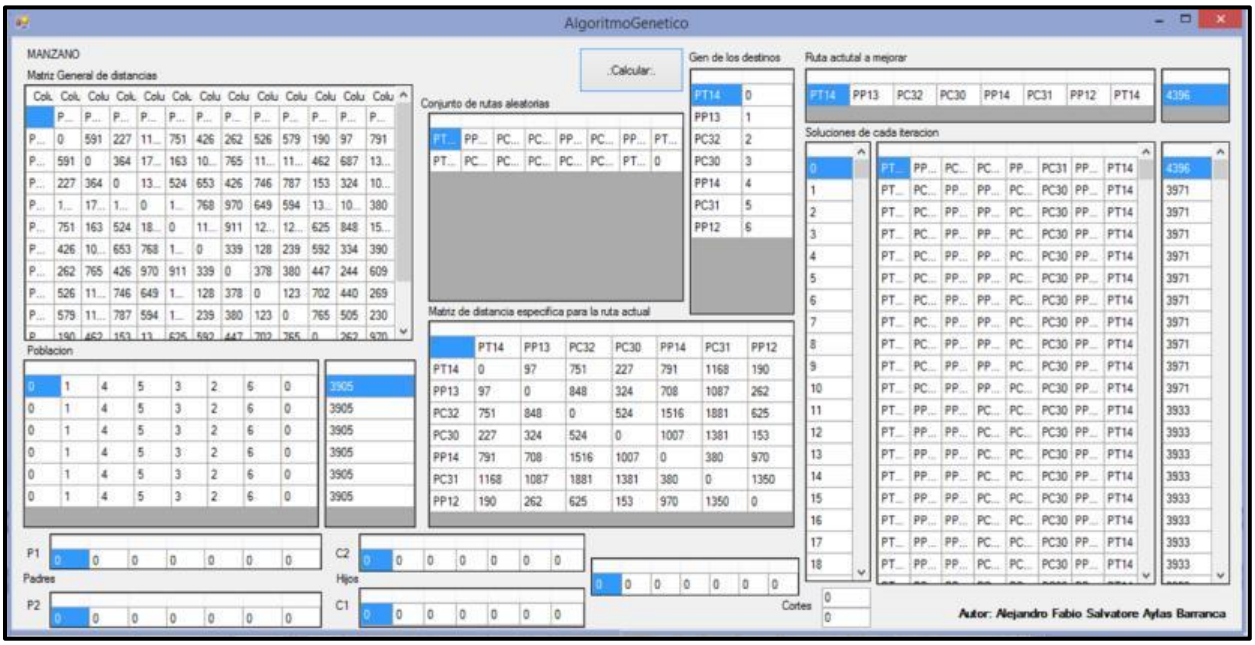
Ventana del formulario AlgoritmoBusquedaTabu



Nota. Se ve las operaciones del algoritmo. Los códigos de programación se encuentran en el Anexo 10.6.

Figura 11.9

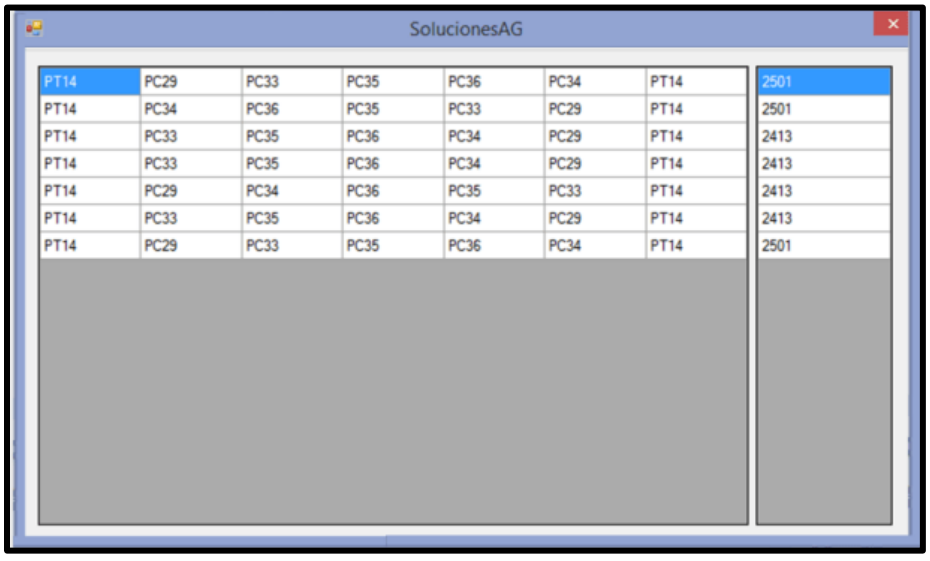
Ventana del formulario AlgoritmoGenetico



Nota. Se ve las operaciones del algoritmo. Los códigos de programación se encuentran en el Anexo 10.7.

Figura 11.10

Ventana del formulario SolucionesAG



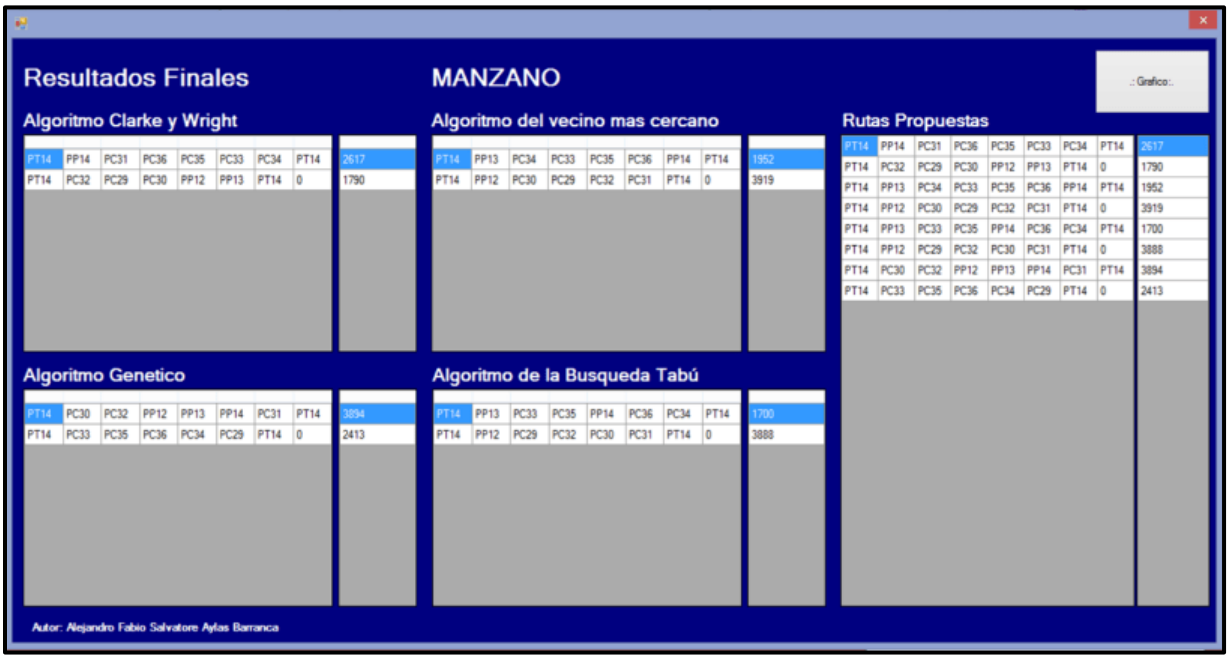
Nota. En cada fila se muestra la mejor solución de cada corrida del algoritmo genético. Los códigos de programación se encuentran en el Anexo 10.8.

Los primeros algoritmos en ejecutarse son de las heurísticas. El primer formulario en efectuar es del algoritmo de Clarke y Wright (Figura 11.6), una vez finalizado sus resultados son almacenados en el formulario *ResultadosFinales*, luego se desarrolla el formulario del algoritmo del Vecino más Cercano (Figura 11.7), los resultados son de la misma manera almacenados en el formulario *ResultadosFinales*. Los dos últimos algoritmos en ejecutarse son las metaheurísticas. El primero es el formulario del algoritmo de la Búsqueda Tabú (Figura 11.8), este comienza recogiendo los resultados del algoritmo anterior (el algoritmo del Vecino más Cercano), para realizar quince iteraciones a cada uno de ellos, luego la mejor solución es seleccionado y mandado al formulario *ResultadosFinales*. Finalizado el proceso anterior, comienza la ejecución del formulario del algoritmo genético (Figura 11.9), este último comienza realizando los pasos previos mencionados en el ejemplo de la sección V.2. (creación aleatoria de soluciones iniciales), luego para cada solución inicial o ruta se realiza diez corridas con cuatrocientas iteraciones en cada una de ellas , el mejor resultado en cada corrida se va almacenando en el formulario *SolucionesAG* (Figura 11.10), la mejor solución de todas las corridas es seleccionada y mandada al formulario *ResultadosFinales*.

Cuando se hayan concluido todos los algoritmos, se va a visualizar en el formulario de *ResultadosFinales* todas las soluciones (Figura 11.11).

Figura 11.11

Ventana del formulario ResultadosFinales



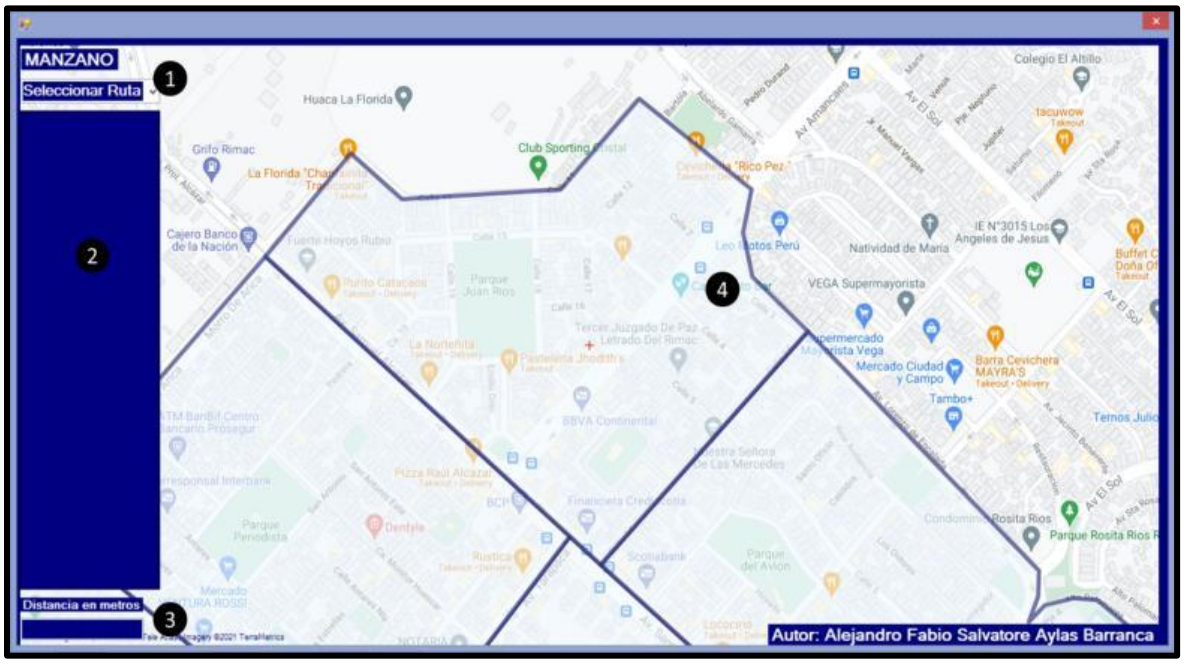
Nota. Los códigos de programación se encuentran en Anexo 10.9.

Las rutas propuestas son en conjunto, cada uno de los resultados obtenidos por las heurísticas y metaheurísticas porque generalmente sus soluciones son distintas entre ellas. Sin embargo, habrá excepciones donde se presente que algunas soluciones que coincidan en la distancia, debido a ello el aplicativo va a considerar una solución de todas las similares, además de que descarta a aquellos resultados que sean rutas con menos de cuatro destinos a visitar.

De la Figura 11.11 cuando es ejecutado el botón “Grafico”, va a aparecer el ultimo formulario del aplicativo llamado *MostrarRutas*, el propósito es ver la trayectoria referencial de cada ruta obtenida por los algoritmos. El formulario se ve en la Figura 11.12.

Figura 11.12

Ventana del formulario MostrarRutas



Nota. Los puntos negros indican: en (1) la lista del total de rutas, en (2) se va a mostrar una matriz con la secuencia de destinos según sea la ruta seleccionada, en (3) se va a mostrar la distancia en metros de la misma ruta y en (4) un mapa para ver la secuencia de la ruta seleccionada. Los códigos de programación se encuentran en el Anexo 10.10.

Para visualizar la trayectoria de cada ruta de la Figura 11.12 se debe seleccionar un elemento de la lista de rutas como se ve en la Figura 11.13.

Figura 11.13

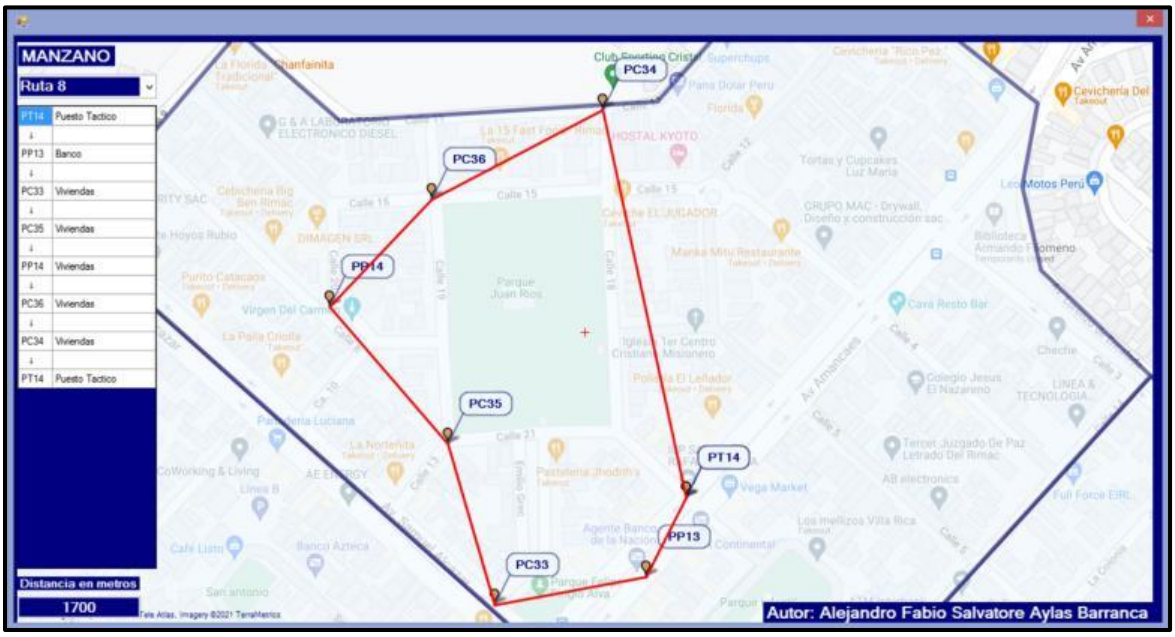
Ventana del formulario MostrarRutas



Cuando se haya elegido alguna de las opciones vistas en la Figura 11.13, se mostrará la trayectoria seleccionada como se ve en la Figura 11.14.

Figura 11.14

Ventana del formulario MostrarRutas



Nota. La trayectoria mostrada es la ruta 8.

Como se mencionó en las bases teóricas los procedimientos de estos algoritmos son complejos, por ello consumen altos recursos computacionales para brindar resultados. No se presentaron limitaciones tecnológicas en lo referido en la ejecución de los algortimos matemáticos en el aplicativo desktop, debido que se utilizó una Laptop con las siguientes características:

- Windows 8.1 Pro
- Procesador: Intel ® Core ™ i3-3110M CPU 240GHz
- Memoria instalada (RAM): 8.00 GB (7.85 GB utilizable)
- Tipo de sistema: Sistema operativo de 64 bits, procesador x64

5.4. Análisis de resultados

El análisis consistió en comparar la distancia de las rutas empíricas (construidos según lo expuesto en la sección V.1) con las rutas que son resultados de los algoritmos matemáticos. Se han planteado escenarios que puedan ocurrir durante las rutinas de patrullaje, luego mediante cuadros comparativos se buscó evidenciar que los algortimos matemáticos logran diseñar rutas con recorridos cortos, siendo mejores a las rutas empíricas.

Se van a presentar rutas ideales para una hora de patrullaje móvil. Además, si bien es cierto, en algunas situaciones un vehículo es asignado a patrullar dos sectores, se va desarrollar el caso en que se asigne a custodiar un sector.

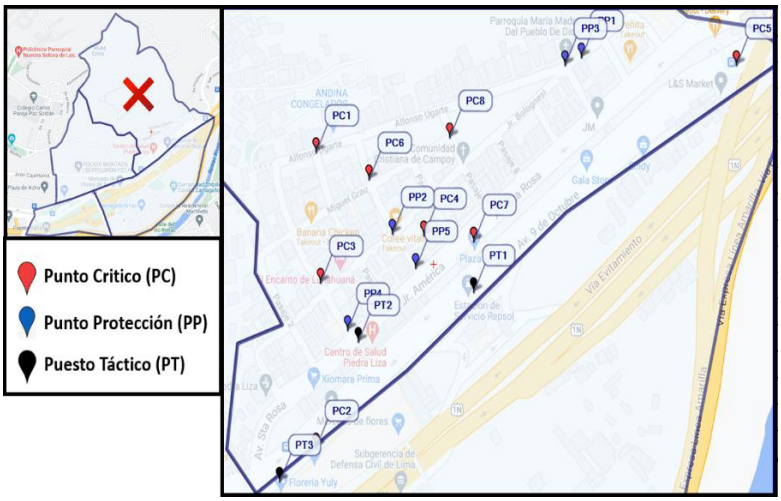
Es pertinente mencionar que la actividad de patrullaje móvil no se realiza en todos los sectores, algunos escenarios que justifican lo dicho son: por decisión de las autoridades y espacios geográficos que dificultan el desplazamiento de los vehículos. La actividad común en estos casos es el patrullaje táctico.

5.4.1. Análisis de resultados para CPNP Piedra Liza

Análisis para el sector S1-1

Figura 12

Mapa del sector S1-1 de CPNP Piedra Liza



Nota. La leyenda se encuentra en Anexo 9.1.

La Tabla 12 indica los puntos del sector.

Tabla 17

Información del sector S1-1 de CPNP Piedra Liza

		Puntos del sector							
Origen	Táctico Índice	PT1 1	PT2 2	PT3 3					
Destinos	Crítico Índice:	PC1 1	PC2 2	PC3 3	PC4 4	PC5 5	PC6 6	PC7 7	PC8 8
	Protección Índice	PP1 9	PP2 10	PP3 11	PP4 12	PP5 13			

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

Con los datos de la Tabla 12 se han seleccionado aleatoriamente los puntos para construir rutas empíricas.

Tabla 12.1

Datos aleatorios para construir rutas empíricas en el S1-1 de CPNP Piedra Liza

Escenario 1 : 100% (total Puntos) : 13 puntos													
O	Índice:	2											
	Punto Inicial:	PT2											
D	Aleatorios:	0.619	0.79	0.64	0.825	0.056	0.425	0.683	0.142	0.473	0.798	0.51	0.616	0.732
	Índice:	7	3	6	1	13	11	5	12	10	2	9	8	4
	Destino:	PC7	PC3	PC6	PC1	PP5	PP3	PC5	PP4	PP2	PC2	PP1	PC8	PC4
Escenario 2 : 80% (total Puntos) : 10 puntos													
O	Índice:	3											
	Punto Inicial:	PT3											
D	Aleatorios:	0.939	0.579	0.2	0.582	0.541	0.15	0.632	0.589	0.698	0.695	0.962	0.279	0.846
	Índice:	2	9	12	8	10	13	6	7	4	5			
	Destino:	PC2	PP1	PP4	PC8	PP2	PP5	PC6	PC7	PC4	PC5			
Escenario 3 : 60% (total Puntos) : 8 puntos													
O	Índice:	1											
	Punto Inicial:	PT1											
D	Aleatorios:	0.924	0.212	0.66	0.334	0.21	0.914	0.902	0.617	0.052	0.279	0.441	0.131	0.934
	Índice:	2	10	5	8	11	3	4	6					
	Destino:	PC2	PP2	PC5	PC8	PP3	PC3	PC4	PC6					
Escenario 4 : 40% (total Puntos) : 5 puntos													
O	Índice:	1											
	Punto Inicial:	PT1											
D	Aleatorios:	0.648	0.025	0.356	0.342	0.418	0.357	0.262	0.201	0.163	0.919	0.259	0.52	0.162
	Índice:	2	13	6	7	4								
	Destino:	PC2	PP5	PC6	PC7	PC4								

Nota. Se muestra los resultados aleatorios de las fórmulas de Excel. Se denota que O es origen y D es destinos.

Según la Tabla 12.1, fue posible construir cuatro rutas empíricas distintas. En el primer, segundo, tercer y cuarto escenario se visitan trece, diez, ocho y cinco puntos respectivamente.

Los datos acerca del punto de partida y los destinos de escenario han sido usados para presentar los resultados de los algoritmos matemáticos en la Tabla 12.2.

Tabla 12.2

Resultados de la aplicación de algoritmos matemáticos

Comparación de Resultados : Escenario 1		
Ruta empírica	Ruta:	PT2 PC7 PC3 PC6 PC1 PP5 PP3 PC5 PP4 PP2 PC2 PP1 PC8 PC4 PT2
	Distancia:	5449 metros
A. Clarke y Wright	Ruta:	PT2 PP1 PC5 PP3 PC7 PC8 PC4 PP5 PP2 PC6 PC1 PC3 PC2 PP4 PT2
	Distancia:	2761 metros
A. Vecino más Cercano	Ruta:	PT2 PP4 PC3 PC1 PC2 PC6 PP2 PP5 PC4 PC8 PC7 PP3 PP1 PC5 PT2
	Distancia:	2846 metros
A. Búsqueda Tabú	Ruta:	PT2 PP4 PC2 PC3 PC1 PC6 PP2 PC8 PP3 PP1 PC5 PC7 PC4 PP5 PT2
	Distancia:	2711 metros
A. Genético	Ruta:	PT2 PP4 PC2 PC3 PC1 PP2 PP5 PC7 PC5 PP1 PP3 PC8 PC4 PC6 PT2
	Distancia:	2772 metros
Comparación de Resultados : Escenario 2		
Ruta empírica	Ruta:	PT3 PC2 PP1 PP4 PC8 PP2 PP5 PC6 PC7 PC4 PC5 PT3
	Distancia:	5126 metros
A. Clarke y Wright	Ruta:	PT3 PP1 PC5 PC7 PC8 PC4 PP5 PP2 PC6 PP4 PC2 PT3
	Distancia:	2859 metros
A. Vecino más Cercano	Ruta:	PT3 PC2 PP4 PC6 PP2 PP5 PC4 PC8 PC7 PP1 PC5 PT3
	Distancia:	2859 metros
A. Búsqueda Tabú	Ruta:	PT3 PC2 PP4 PP5 PC4 PC7 PC5 PP1 PC8 PP2 PC6 PT3
	Distancia:	2813 metros
A. Genético	Ruta:	PT3 PC2 PP5 PC4 PC8 PP1 PC5 PC7 PP2 PC6 PP4 PT3
	Distancia:	2825 metros

Comparación de Resultados : Escenario 3		
Ruta empírica	Ruta:	PT1 PC2 PP2 PC5 PC8 PP3 PC3 PC4 PC6 PT1
	Distancia:	4481 metros
A. Clarke y Wright	Ruta:	PT1 PC3 PC2 PC6 PP2 PC4 PC8 PP3 PC5 PT1
	Distancia:	2663 metros
A. Vecino más Cercano	Ruta:	PT1 PC8 PC4 PP2 PC6 PC3 PC2 PP3 PC5 PT1
	Distancia:	2681 metros
A. Búsqueda Tabú	Ruta:	PT1 PC2 PC3 PC6 PP2 PC4 PC8 PP3 PC5 PT1
	Distancia:	2625 metros
A. Genético	Ruta:	PT1 PC5 PP3 PC8 PC6 PC3 PC2 PP2 PC4 PT1
	Distancia:	2627 metros
Comparación de Resultados : Escenario 4		
Ruta empírica	Ruta:	PT1 PC2 PP5 PC6 PC7 PC4 PT1
	Distancia:	1544 metros
A. Clarke y Wright	Ruta:	PT1 PC6 PC2 PP5 PC4 PC7 PT1
	Distancia:	1045 metros
A. Vecino más Cercano	Ruta:	PT1 PC7 PC4 PP5 PC6 PC2 PT1
	Distancia:	1038 metros
A. Búsqueda Tabú	Ruta:	PT1 PC7 PC4 PC6 PC2 PP5 PT1
	Distancia:	1037 metros
A. Genético	Ruta:	PT1 PC7 PC4 PC6 PC2 PP5 PT1
	Distancia:	1037 metros

Los resultados expuestos en la Tabla 12.2 muestran que las rutas ofrecidas por los cuatro algoritmos matemáticos son mejores porque las distancias del recorrido son menores en comparación a las trayectorias decididas empíricamente. La comparación porcentual se ve en Tabla 12.3.

Tabla 12.3

Comparación porcentual de los resultados

	Escenario 1		Escenario 2		Escenario 3		Escenario 4	
	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual
Ruta empírica	5449	---	5126	---	4481	---	1544	---
A. Clarke y Wright	2761	49.3%	2859	44.2%	2663	40.6%	1045	32.3%
A. de Vecino más Cercano	2846	47.8%	2859	44.2%	2681	40.2%	1038	32.8%
A. de la Búsqueda Tabú	2711	50.2%	2813	45.1%	2625	41.4%	1037	32.8%
A. Genético	2772	49.1%	2825	44.9%	2627	41.4%	1037	32.8%

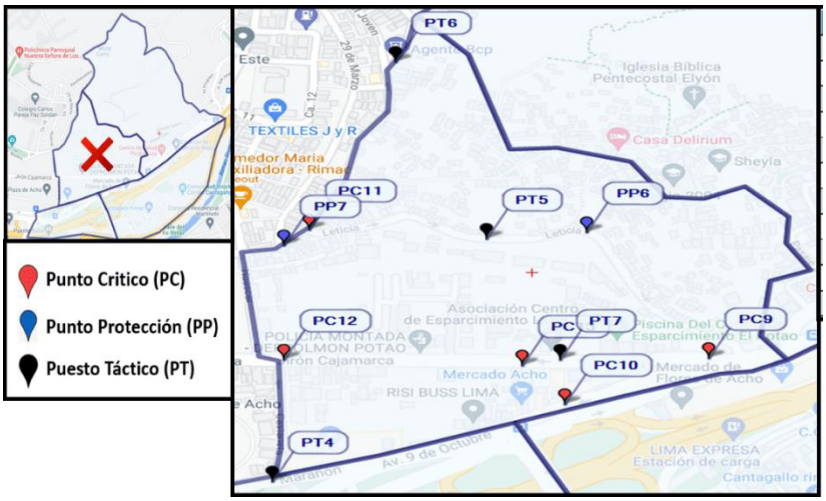
Para los dos primeros escenarios, estas son decisiones menos frecuentes durante el patrullaje, porque realizar desplazamientos que involucren visitar más de diez puntos en una sola trayectoria es muy extenso, aun así, los algoritmos matemáticos lograron reducir en promedio 49% y 44% respectivamente las distancias de recorrido de las rutas empíricas del primer y segundo escenario respectivamente.

Con respecto a los dos últimos escenarios, la decisión de visitar de cinco a nueve puntos si es común en el patrullaje, de la misma manera que en el caso anterior, los algoritmos matemáticos para el tercer y cuarto escenario logran reducir en promedio 41% y 33% la distancia de las rutas empíricas.

Análisis para el sector S1-2

Figura 13

Mapa del sector S1-2 de CPNP Piedra Liza



Nota. La leyenda se encuentra en Anexo 9.1.

La información mostrada en la Tabla 13 indica que existen siete puntos a custodiar y cuatro puestos tácticos. En este sector se hace patrullaje táctico, la decisión se debe a que geográficamente solo hay dos avenidas por donde puede desplazarse el vehículo y ahí están ubicados la mayoría de los puntos a vigilar, un mejor control es que el vehículo permanezca situado cerca a esos puntos.

Tabla 13

Puntos de patrullaje del sector S1-2 de CPNP Piedra Liza

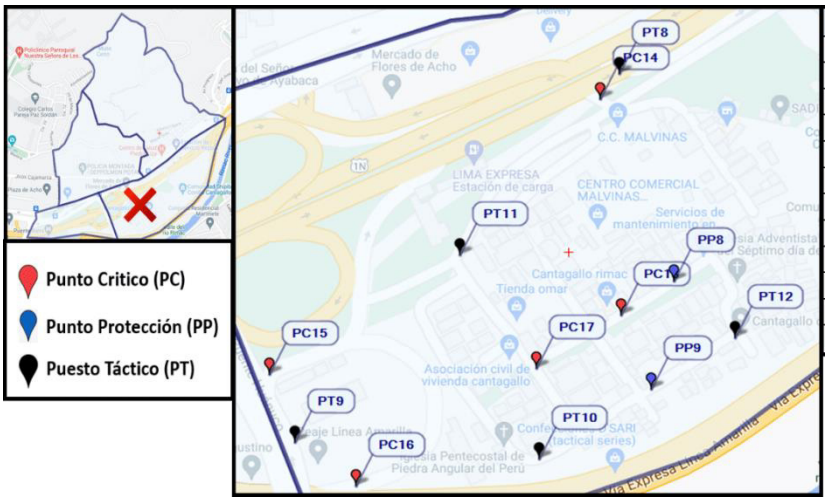
		Puntos del sector				
Origen	Táctico	PT4	PT5	PT6	PT7	
	Índice	1	2	3	4	
Destinos	Crítico	PC9	PC10	PC11	PC12	PC13
	Índice	1	2	3	4	5
	Protección	PP6	PP7			
	Índice	6	7			

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

Análisis para el sector S2-1

Figura 14

Mapa del sector S2-1 de CPNP Piedra Liza



Nota. La leyenda se encuentra en Anexo 9.1.

La Tabla 14 indica que existen siete puntos que son regularmente visitados durante la rutina de patrullaje, se añade también que se utilizan cinco puestos tácticos para el inicio del recorrido.

Tabla 14

Puntos de patrullaje del sector S2-1 de CPNP Piedra Liza

		Puntos del sector				
Origen	Táctico	PT8	PT9	PT10	PT11	PT12
	Índice	1	2	3	4	5
Destinos	Crítico	PC14	PC15	PC16	PC17	PC18
	Índice	1	2	3	4	5
	Protección	PP8	PP9			
	Índice	6	7			

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

Con los datos de la Tabla 14 se han seleccionado aleatoriamente puntos para construir rutas empíricas. La selección está en la Tabla 14.1.

Tabla 14.1

Datos aleatorios para construir rutas empíricas en el S2-1 de CPNP Piedra Liza

Escenario 1 : 100% (total Puntos) : 7 puntos								
O	Índice:	5						
	Punto Inicial:	PT12						
D	Aleatorios:	0.581	0.967	0.519	0.545	0.422	0.808	0.783
	Índice:	4	1	6	5	7	2	3
	Destino:	PC17	PC14	PP8	PC18	PP9	PC15	PC16
Escenario 2 : 80% (total Puntos) : 6 puntos								
O	Índice:	3						
	Punto Inicial:	PT10						
D	Aleatorios:	0.731	0.78	0.243	0.231	0.938	0.343	0.088
	Índice:	3	2	5	6	1	4	
	Destino:	PC16	PC15	PC18	PP8	PC14	PC17	
Escenario 3 : 60% (total Puntos) : 4 puntos								
O	Índice:	4						
	Punto Inicial:	PT11						
D	Aleatorios:	0.302	0.486	0.361	0.681	0.754	0.597	0.238
	Índice:	6	4	5	2			
	Destino:	PP8	PC17	PC18	PC15			

Nota. Se muestra los resultados aleatorios de las fórmulas de Excel. Se denota que O es origen y D es destinos.

Según la Tabla 14.1, fue posible construir tres rutas empíricas distintas. En el primer, segundo y tercer escenario se visitan siete, seis y cuatro puntos respectivamente.

No se planteó un cuarto escenario debido a que tendría como resultado construir una ruta en donde se visite menos de cuatro puntos, esta decisión no se daría durante el patrullaje porque no se cubriría un número de puntos adecuado.

El punto de partida y los destinos de cada escenario son usados para presentar los resultados de en la Tabla 14.2.

Tabla 14.2

Resultados de la aplicación de algoritmos matemáticos

Comparación de Resultados : Escenario 1										
Ruta empírica	Ruta:	PT12	PC17	PC14	PP8	PC18	PP9	PC15	PC16	PT12
	Distancia:	2701	metros							
A. Clarke y Wright	Ruta:	PT12	PC16	PC15	PC17	PC14	PC18	PP9	PP8	PT12
	Distancia:	2079	metros							
A. Vecino más Cercano	Ruta:	PT12	PP8	PP9	PC18	PC14	PC17	PC16	PC15	PT12
	Distancia:	2080	metros							
A. Búsqueda Tabú	Ruta:	PT12	PP8	PC14	PC15	PC16	PC17	PC18	PP9	PT12
	Distancia:	1977	metros							
A. Genético	Ruta:	PT12	PP8	PC14	PC15	PC16	PC17	PC18	PP9	PT12
	Distancia:	1977	metros							
Comparación de Resultados : Escenario 2										
Ruta empírica	Ruta:	PT10	PC17	PC14	PC15	PC16	PC18	PP8	PT10	
	Distancia:	2056	metros							
A. Clarke y Wright	Ruta:	PT10	PC16	PC15	PC14	PP8	PC18	PC17	PT10	
	Distancia:	1759	metros							
A. Vecino más Cercano	Ruta:	PT10	PC17	PC18	PP8	PC14	PC16	PC15	PT10	
	Distancia:	1780	metros							
A. Búsqueda Tabú	Ruta:	PT10	PC17	PC18	PP8	PC14	PC15	PC16	PT10	
	Distancia:	1759	metros							
A. Genético	Ruta:	PT10	PC16	PC15	PC17	PC14	PP8	PC18	PT10	
	Distancia:	1758	metros							

Comparación de Resultados : Escenario 3							
Ruta empírica	Ruta:	PT11	PP8	PC17	PC18	PC15	PT11
	Distancia:	2021	metros				
A. Clarke y Wright	Ruta:	PT11	PP8	PC18	PC17	PC15	PT11
	Distancia:	1668	metros				
A. Vecino más Cercano	Ruta:	PT11	PC17	PC18	PP8	PC15	PT11
	Distancia:	1679	metros				
A. Búsqueda Tabú	Ruta:	PT11	PP8	PC18	PC17	PC15	PT11
	Distancia:	1668	metros				
A. Genético	Ruta:	PT11	PP8	PC18	PC17	PC15	PT11
	Distancia:	1668	metros				

Las soluciones de los algoritmos son mejores, las distancias de recorrido son menores. Ver en la Tabla 14.3.

Tabla 14.3

Comparación porcentual de los resultados

	Escenario 1		Escenario 2		Escenario 3	
	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual
Ruta empírica	2701	---	2056	---	2021	---
A. Clarke y Wright	2079	23%	1759	14.4%	1668	17.5%
A. Vecino más Cercano	2080	23%	1780	13.4%	1679	16.9%
A. Búsqueda Tabú	1977	26.8%	1759	14.4%	1668	17.5%
A. Genético	1977	26.8%	1758	14.5%	1668	17.5%

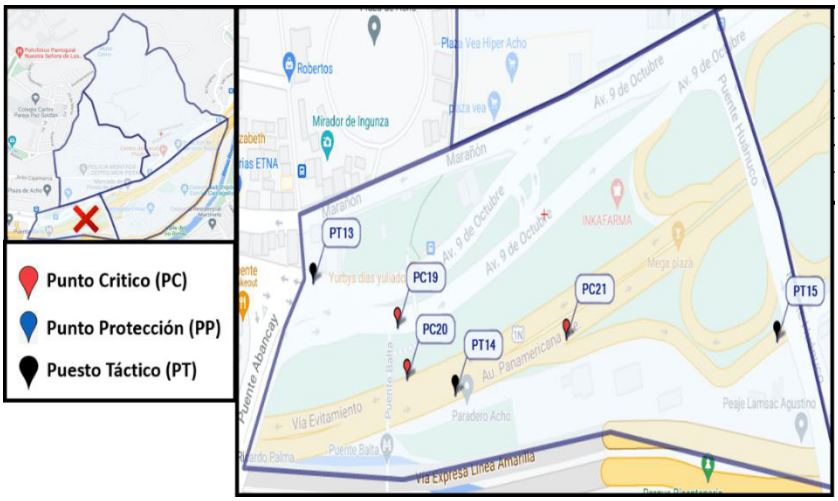
Los dos primeros escenarios planteados son decisiones frecuentes durante el patrullaje porque cotidianamente se recorren de cinco a nueve destinos, los algoritmos logran reducir en promedio 24% y 14% las distancias de recorrido en las rutas construidas empíricamente en el primer y segundo escenario respectivamente.

Con respecto al último escenario, este es un recorrido corto y son decisiones no cotidianas porque implica visitar una pequeña cantidad de puntos, aun así, la aplicación de los algoritmos reduce en promedio 17% la distancia de recorrido.

Análisis para el sector S2-2

Figura 15

Mapa del sector S2-2 de CPNP Piedra Liza



Nota. La leyenda se encuentra en Anexo 9.1.

La información mostrada en la Tabla 15 indica que existen tres puntos a custodiar y tres puestos tácticos. En este sector se realiza el patrullaje táctico, por ser un sector con una cantidad de puntos críticos mínima. El vehículo se estaciona en alrededores de los puntos críticos además de los puestos tácticos.

Tabla 15

Información del sector S2-2 de CPNP Piedra Liza

Puntos del sector				
Origen	Táctico	PT13	PT14	PT15
	Índice	1	2	3
Destinos	Crítico	PC19	PC20	PC21
	Índice	1	2	3

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

En la Tabla 16 se compara las distancias de las rutas para patrullaje cuando se toma una decisión empírica y cuando se toma una decisión en base a técnicas matemáticas.

Tabla 16

Resultados finales para la CPNP Piedra Liza

Sectores	Escenarios	Distancia de la ruta empírica (metros)	Promedio de distancias de rutas con Heurísticas y Metaheurísticas (metros)	Porcentaje de reducción	Estado
S1 - 1	Escenario 1	5449	2772.5	49.1%	Reducción
	Escenario 2	5126	2839	44.6%	Reducción
	Escenario 3	4481	2649	40.9%	Reducción
	Escenario 4	1544	1039.25	32.7%	Reducción
S1 - 2	No se realiza patrullaje móvil				
S2 - 1	Escenario 1	2701	2028.25	24.91%	Reducción
	Escenario 2	2056	1764	14.20%	Reducción
	Escenario 3	2021	1670.75	17.33%	Reducción
S2 - 2	No se realiza patrullaje móvil				

Nota. La ultima columna puntualiza el efecto de usar técnicas matemáticas sobre decisiones empíricas.

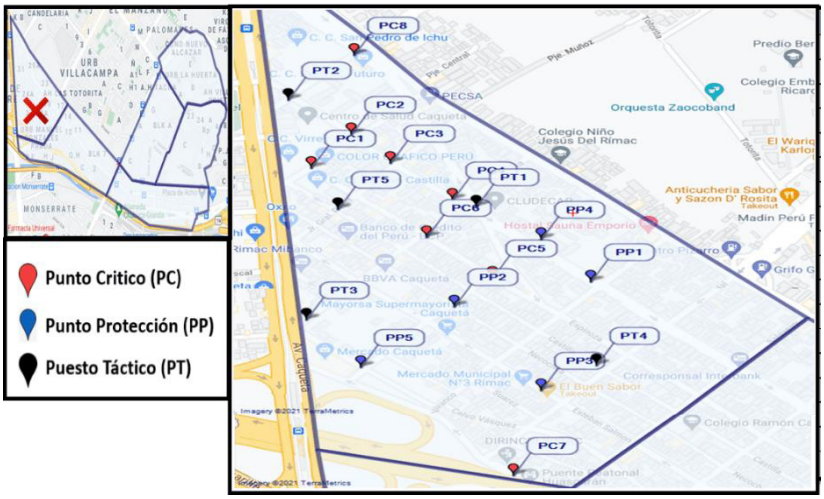
Según la Tabla 16 se puede observar que la aplicación de heurísticas y metaheurísticas para diseñar rutas, han logrado reducir la distancia del recorrido de los vehículos en las rutinas de patrullaje móvil para la comisaría PNP Piedra Liza.

5.4.2. Análisis de resultados para CPNP Rímac

Análisis para el sector S1

Figura 16

Mapa del sector S1 de CPNP Rímac



Nota. La leyenda se encuentra en Anexo 9.2.

En la Tabla 17 se expone que existen trece puntos que son considerados a visitar durante el patrullaje, además se utilizan cinco puestos tácticos para el inicio del recorrido.

Tabla 17

Información del sector S1 de CPNP Rímac

		Puntos del sector				
Origen	Táctico Índice	PT1 1	PT2 2	PT3 3	PT4 4	PT5 5
Destinos	Crítico Índice:	PC1 1	PC2 2	PC3 3	PC4 4	
	Crítico Índice:	PC5 5	PC6 6	PC7 7	PC8 8	
	Protección Índice	PP1 9	PP2 10	PP3 11	PP4 12	PP5 13

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

De los datos de la Tabla 17 se han seleccionado aleatoriamente los puntos para construir rutas empíricas. Se muestra en la Tabla 17.1.

De lo mostrado en la Tabla 17.1, se construyeron cuatro rutas empíricas. En el primer, segundo, tercer y cuarto escenario se visitan trece, diez, ocho y cinco puntos respectivamente.

Tabla 17.1

Datos aleatorios para construir rutas empíricas en el S1 de CPNP Rímac

Escenario 1 : 100% (total Puntos) : 13 puntos													
O	Índice:	4											
	Punto Inicial:	PT4											
D	Aleatorios:	0.018	0.717	0.274	0.934	0.121	0.212	0.506	0.588	0.467	0.778	0.452	0.338	0.65
	Índice:	13	3	10	1	12	11	6	5	7	2	8	9	4
	Destino:	PP5	PC3	PP2	PC1	PP4	PP3	PC6	PC5	PC7	PC2	PC8	PP1	PC4
Escenario 2 : 80% (total Puntos) : 10 puntos													
O	Índice:	2											
	Punto Inicial:	PT2											
D	Aleatorios:	0.072	0.14	0.454	0.322	0.531	0.763	0.438	0.346	0.411	0.48	0.73	0.261	0.27
	Índice:	13	12	5	9	3	1	6	8	7	4			
	Destino:	PP5	PP4	PC5	PP1	PC3	PC1	PC6	PC8	PC7	PC4			
Escenario 3 : 60% (total Puntos) : 8 puntos													
O	Índice:	5											
	Punto Inicial:	PT5											
D	Aleatorios:	0.791	0.768	0.036	0.339	0.084	0.363	0.64	0.81	0.298	0.19	0.784	0.48	0.555
	Índice:	2	4	13	9	12	8	5	1					
	Destino:	PC2	PC4	PP5	PP1	PP4	PC8	PC5	PC1					
Escenario 4 : 40% (total Puntos) : 5 puntos													
O	Índice:	2											
	Punto Inicial:	PT2											
D	Aleatorios:	0.02	0.013	0.176	0.485	0.094	0.028	0.652	0.572	0.446	0.76	0.057	0.631	0.303
	Índice:	12	13	8	5	9								
	Destino:	PP4	PP5	PC8	PC5	PP1								

Nota. Se muestra los resultados aleatorios de las fórmulas de Excel. Se denota que O es origen y D es destinos.

Los puntos seleccionados han sido usados para presentar los resultados de los algoritmos en la Tabla 17.2.

Tabla 17.2

Resultados de la aplicación de algoritmos matemáticos

Comparación de Resultados : Escenario 1																
Ruta empírica	Ruta:	PT4	PP5	PC3	PP2	PC1	PP4	PP3	PC6	PC5	PC7	PC2	PC8	PP1	PC4	PT4
	Distancia:	7274	metros													
A. Clarke y Wright	Ruta:	PT4	PC2	PC1	PC8	PC3	PP5	PC6	PC4	PP2	PC5	PC7	PP3	PP4	PP1	PT4
	Distancia:	3558	metros													
A. Vecino más Cercano	Ruta:	PT4	PP1	PP4	PC5	PP2	PC6	PC4	PC3	PC2	PC8	PC1	PP5	PC7	PP3	PT4
	Distancia:	2980	metros													
A. Búsqueda Tabú	Ruta:	PT4	PP1	PP4	PC5	PC4	PC6	PC3	PC8	PC2	PC1	PP5	PP2	PC7	PP3	PT4
	Distancia:	2857	metros													
A. Genético	Ruta:	PT4	PP3	PC7	PC5	PP2	PP5	PC1	PC2	PC8	PC3	PC6	PC4	PP4	PP1	PT4
	Distancia:	2929	metros													
Comparación de Resultados : Escenario 2																
Ruta empírica	Ruta:	PT2	PP5	PP4	PC5	PP1	PC3	PC1	PC6	PC8	PC7	PC4	PT2			
	Distancia:	5646	metros													
A. Clarke y Wright	Ruta:	PT2	PP4	PP1	PC7	PC5	PC4	PC6	PC3	PP5	PC8	PC1	PT2			
	Distancia:	3415	metros													
A. Vecino más Cercano	Ruta:	PT2	PC1	PC8	PC3	PC6	PC4	PC5	PP4	PP1	PC7	PP5	PT2			
	Distancia:	2895	metros													
A. Búsqueda Tabú	Ruta:	PT2	PC1	PP5	PC7	PP1	PP4	PC5	PC4	PC6	PC3	PC8	PT2			
	Distancia:	2810	metros													
A. Genético	Ruta:	PT2	PC1	PP5	PC4	PC5	PP4	PP1	PC7	PC6	PC3	PC8	PT2			
	Distancia:	2966	metros													
Comparación de Resultados : Escenario 3																
Ruta empírica:	Ruta:	PT5	PC2	PC4	PP5	PP1	PP4	PC8	PC5	PC1	PT5					
	Distancia:	4293	metros													
A. Clarke y Wright	Ruta:	PT5	PP4	PP1	PC5	PC4	PP5	PC2	PC8	PC1	PT5					
	Distancia:	2757	metros													
A. Vecino más Cercano	Ruta:	PT5	PC2	PC8	PC1	PP5	PC4	PC5	PP4	PP1	PT5					
	Distancia:	2776	metros													

Comparación de Resultados : Escenario 3											
A. Búsqueda Tabú	Ruta:	PT5	PP5	PC5	PP1	PP4	PC4	PC8	PC2	PC1	PT5
	Distancia:	2515	metros								
A. Genético	Ruta:	PT5	PP5	PC5	PP1	PP4	PC4	PC8	PC2	PC1	PT5
	Distancia:	2515	metros								
Comparación de Resultados : Escenario 4											
Ruta empírica	Ruta:	PT2	PP4	PP5	PC8	PC5	PP1	PT2			
	Distancia:	4367	metros								
A. Clarke y Wright	Ruta:	PT2	PP4	PP1	PC5	PP5	PC8	PT2			
	Distancia:	2810	metros								
A.Vecino más Cercano	Ruta:	PT2	PC8	PP5	PC5	PP4	PP1	PT2			
	Distancia:	2824	metros								
A. Búsqueda Tabú	Ruta:	PT2	PC8	PP4	PP1	PC5	PP5	PT2			
	Distancia:	2612	metros								
A. Genético	Ruta:	PT2	PC8	PP4	PP1	PC5	PP5	PT2			
	Distancia:	2612	metros								

Se ve en la Tabla 17.2 que las rutas diseñadas con algoritmos matemáticos son mejores. Ver la Tabla 17.3.

Tabla 17.3

Comparación porcentual de los resultados

	Escenario 1		Escenario 2		Escenario 3		Escenario 4	
	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual
Ruta empírica	7274	---	5646	---	4293	---	4367	---
A. Clarke y Wright	3558	51.1%	3415	39.5%	2757	35.8%	2810	35.7%
A. Vecino más Cercano	2980	59%	2895	48.7%	2776	35.3%	2824	35.3%
A. Búsqueda Tabú	2857	60.7%	2810	50.2%	2515	41.4%	2612	40.2%
A. Genético	2929	59.7%	2966	47.5%	2515	41.4%	2612	40.2%

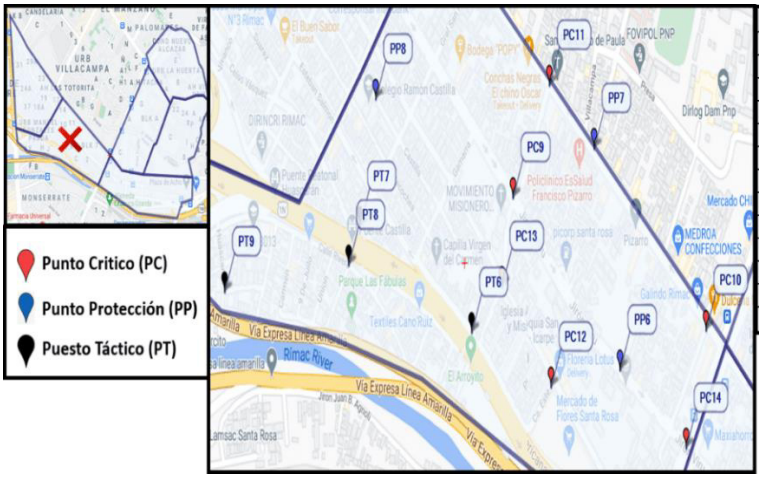
Los dos escenarios iniciales plantean rutas con más de diez puntos, esta decisión no es frecuente, los algoritmos matemáticos han reducido en promedio 57% y 46% las distancias en el primer y segundo escenario respectivamente.

Los dos últimos escenarios son decisiones cotidianas, esto se debe a que visitar de cinco a nueve puntos son elecciones que se dan día a día, el uso de los algoritmos matemáticos logra reducir en promedio 38% y 37% las distancias de recorrido en las rutas empíricas construidas en el tercer y cuarto escenario respectivamente.

Análisis para el sector S2

Figura 17

Mapa del sector S2 de CPNP Rímac



Nota. La leyenda se encuentra en Anexo 9.2.

La Tabla 18 indica los puntos del sector.

Tabla 18

Información del sector S2 de CPNP Rímac

		Puntos del sector					
Origen	Táctico	PT6	PT7	PT8	PT9		
	Índice	1	2	3	4		
Destinos	Crítico	PC9	PC10	PC11	PC12	PC13	PC14
	Índice	1	2	3	4	5	6
	Protección	PP6	PP7	PP8			
	Índice	7	8	9			

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

De la Tabla 18 se han seleccionado aleatoriamente los puntos para construir rutas empíricas. Ver la Tabla 18.1.

Tabla 18.1

Datos aleatorios para construir rutas empíricas en el sector S2 de CPNP Rímac

Escenario 1 : 100% (total Puntos) : 9 puntos										
O	Índice:	1								
	Punto Inicial:	PT6								
	Aleatorios:	0.018	0.717	0.274	0.934	0.121	0.212	0.506	0.588	0.467
D	Índice:	9	2	6	1	8	7	4	3	5
	Destino:	PP8	PC10	PC14	PC9	PP7	PP6	PC12	PC11	PC13
Escenario 2 : 80% (total Puntos) : 7 puntos										
O	Índice:	1								
	Punto Inicial:	PT6								
	Aleatorios:	0.89	0.874	0.381	0.883	0.615	0.455	0.973	0.387	0.826
D	Índice:	2	4	9	3	6	7	1		
	Destino:	PC10	PC12	PP8	PC11	PC14	PP6	PC9		
Escenario 3 : 60% (total Puntos) : 5 puntos										
O	Índice:	3								
	Punto Inicial:	PT8								
	Aleatorios:	0.791	0.768	0.036	0.339	0.084	0.363	0.64	0.81	0.298
D	Índice:	2	3	9	6	8				
	Destino:	PC10	PC11	PP8	PC14	PP7				
Escenario 4 : 40% (total Puntos) : 4 puntos										
O	Índice:	2								
	Punto Inicial:	PT7								
	Aleatorios:	0.038	0.685	0.527	0.714	0.162	0.681	0.08	0.548	0.297
D	Índice:	9	2	5	1					
	Destino:	PP8	PC10	PC13	PC9					

Nota. Se muestra los resultados aleatorios de las fórmulas de Excel. Se denota que O es origen y D es destinos. Para el primer, segundo, tercer y cuarto escenario se visitan nueve, siete, cinco y cuatro puntos respectivamente. Los datos de cada ruta empírica fueron usados para presentar los resultados de los algoritmos en la Tabla 18.2.

Tabla 18.2

Resultados de la aplicación de algoritmos matemáticos

Comparación de Resultados : Escenario 1												
Ruta empírica	Ruta:	PT6	PP8	PC10	PC14	PC9	PP7	PP6	PC12	PC11	PC13	PT6
	Distancia:	6343	metros									
A. Clarke y Wright	Ruta:	PT6	PC14	PC10	PP6	PP7	PC11	PC12	PC9	PC13	PP8	PT6
	Distancia:	4934	metros									
A. Vecino más Cercano	Ruta:	PT6	PC13	PC9	PC11	PC12	PP7	PP6	PC14	PC10	PP8	PT6
	Distancia:	4934	metros									
A. Búsqueda Tabú	Ruta:	PT6	PC13	PC12	PP6	PC14	PC10	PP7	PC11	PC9	PP8	PT6
	Distancia:	4585	metros									
A. Genético	Ruta:	PT6	PP8	PC9	PC11	PP7	PC10	PC14	PP6	PC12	PC13	PT6
	Distancia:	4585	metros									
Comparación de Resultados : Escenario 2												
Ruta empírica	Ruta:	PT6	PC10	PC12	PP8	PC11	PC14	PP6	PC9	PT6		
	Distancia:	7503	metros									
A. Clarke y Wright	Ruta:	PT6	PC14	PC10	PP6	PC12	PC11	PC9	PP8	PT6		
	Distancia:	4779	metros									
A. Vecino más Cercano	Ruta:	PT6	PC9	PC11	PC12	PP6	PC14	PC10	PP8	PT6		
	Distancia:	4801	metros									
A. Búsqueda Tabú	Ruta:	PT6	PC12	PP6	PC14	PC10	PC11	PC9	PP8	PT6		
	Distancia:	4574	metros									
A. Genético	Ruta:	PT6	PP8	PC9	PC11	PC10	PC14	PP6	PC12	PT6		
	Distancia:	4574	metros									
Comparación de Resultados : Escenario 3												
Ruta empírica	Ruta:	PT8	PC10	PC11	PP8	PC14	PP7	PT8				
	Distancia:	9152	metros									
A. Clarke y Wright	Ruta:	PT8	PC14	PC10	PP7	PC11	PP8	PT8				
	Distancia:	4944	metros									
A. Vecino más Cercano	Ruta:	PT8	PP8	PC11	PP7	PC14	PC10	PT8				
	Distancia:	4969	metros									

Comparación de Resultados : Escenario 3								
A. Búsqueda Tabú	Ruta:	PT8	PP8	PC11	PP7	PC10	PC14	PT8
	Distancia:	4944	metros					
A. Genético	Ruta:	PT8	PC14	PC10	PP7	PC11	PP8	PT8
	Distancia:	4944	metros					
Comparación de Resultados : Escenario 4								
Ruta empírica	Ruta:	PT7	PP8	PC10	PC13	PC9	PT7	
	Distancia:	4925	metros					
A. Clarke y Wright	Ruta:	PT7	PC10	PC9	PC13	PP8	PT7	
	Distancia:	4786	metros					
A. Vecino más Cercano	Ruta:	PT7	PP8	PC13	PC9	PC10	PT7	
	Distancia:	4786	metros					
A. Búsqueda Tabú	Ruta:	PT7	PP8	PC9	PC10	PC13	PT7	
	Distancia:	4750	metros					
A. Genético	Ruta:	PT7	PC13	PC10	PC9	PP8	PT7	
	Distancia:	4750	metros					

Los resultados en la Tabla 18.2 muestran en primera impresión, que los algoritmos matemáticos brindan mejores rutas porque las distancias del recorrido son menores en comparación a las trayectorias decididas empíricamente. La comparación porcentual se ve en Tabla 18.3.

Tabla 18.3

Comparación porcentual de los resultados

	Escenario 1		Escenario 2		Escenario 3		Escenario 4	
	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual
Ruta empírica	6343	---	7503	---	9152	---	4925	---
A. Clarke y Wright	4934	22.2%	4779	36.3%	4944	46%	4786	2.8%
A. Vecino más Cercano	4934	22.2%	4801	36%	4969	45.7%	4786	2.8%
A. Búsqueda Tabú	4585	27.7%	4574	39%	4944	46%	4750	4%
A. Genético	4585	27.7%	4574	39%	4944	46%	4750	4%

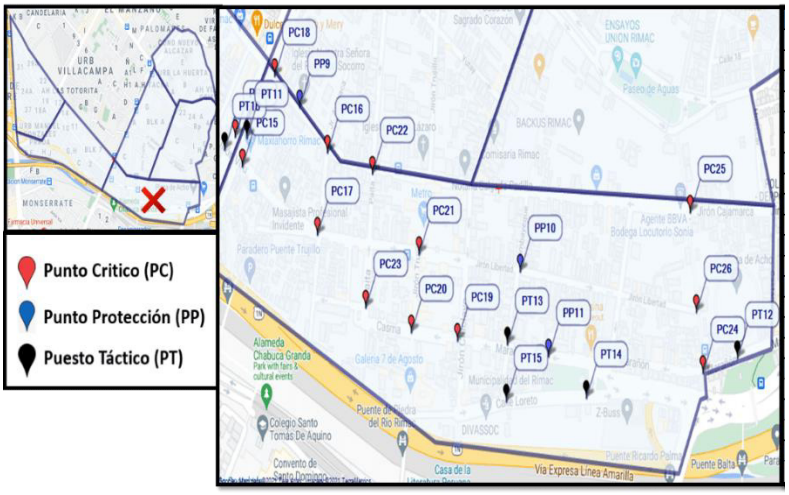
En los tres primeros escenarios, la decisión de visitar de cinco a nueve puntos es común, los algoritmos matemáticos logran reducir en promedio 25 %, 38% y 46% en el primer, segundo y tercer escenario respectivamente las distancias de recorrido de los vehículos en cada ruta empírica.

El último escenario indica visitar al menos cuatro destinos en una sola trayectoria, los algoritmos logran reducir en promedio 3% la distancia para la ruta empírica, no se refleja una diferencia considerable pero se ve una mejora.

Análisis para el sector S3

Figura 18

Mapa del sector S3 de CPNP Rímac



Nota. La leyenda se encuentra en Anexo 9.2.

La información del sector esta en la Tabla 19.

Tabla 19

Información del sector S3 de CPNP Rímac

		Puntos del sector						
Origen	Táctico	PT10	PT11	PT12	PT13	PT14	PT15	
	Índice	1	2	3	4	5	6	
Destinos	Crítico	PC15	PC16	PC17	PC18	PC19	PC20	PC21
	Índice	1	2	3	4	5	6	7
	Crítico	PC22	PC23	PC24	PC25	PC26	PC27	
	Índice	8	9	10	11	12	13	
	Protección	PP9	PP10	PP11				
	Índice	14	15	16				

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

Con los datos de la Tabla 19 se han seleccionado aleatoriamente los puntos para construir rutas empíricas. La selección está en la Tabla 19.1.

Tabla 19.1

Datos aleatorios para construir rutas empíricas en el S3 de CPNP Rímac

Escenario 1 : 100% (total Puntos) : 16 puntos																
O	Índice :	4														
	Punto Inicial:	PT13														
	Aleatorios:	0.27	0.747	0.454	0.351	0.483	0.364	0.793	0.092	0.243	0.845	0.487	0.293	0.9	0.54	0.204	0.406
D	Índice:	13	4	8	11	7	10	3	16	14	2	6	12	1	5	15	9
	Destino:	PC27	PC18	PC22	PC25	PC21	PC24	PC17	PP11	PP9	PC16	PC20	PC26	PC15	PC19	PP10	PC23
Escenario 2 : 80% (total Puntos) : 13 puntos																
O	Índice:	3														
	Punto Inicial:	PT12														
	Aleatorios:	0.089	0.764	0.15	0.083	0.928	0.745	0.923	0.985	0.774	0.444	0.268	0.591	0.5	0.33	0.583	0.19
D	Índice:	15	5	14	16	2	6	3	1	4	10	12	7	9			
	Destino:	PP10	PC19	PP9	PP11	PC16	PC20	PC17	PC15	PC18	PC24	PC26	PC21	PC23			
Escenario 3 : 60% (total Puntos) : 10 puntos																
O	Índice:	1														
	Punto Inicial:	PT10														
	Aleatorios:	0.085	0.983	0.268	0.138	0.959	0.828	0.689	0.29	0.244	0.554	0.311	0.709	0.532	0.655	0.212	0.495
D	Índice:	16	1	12	15	2	3	5	11	13	7						
	Destino:	PP11	PC15	PC26	PP10	PC16	PC17	PC19	PC25	PC27	PC21						

Escenario 4 : 40% (total Puntos) : 6 puntos																	
O	Índice:	5															
	Punto Inicial:	PT14															
	Aleatorios:	0.658	0.973	0.06	0.584	0.295	0.557	0.396	0.921	0.795	0.371	0.87	0.746	0.384	0.727	0.519	0.479
D	Índice:	7	1	16	8	15	9										
	Destino:	PC21	PC15	PP11	PC22	PP10	PC23										

Nota. Se muestra los resultados aleatorios de las fórmulas de Excel. Se denota que O es origen y D es destinos.

Según la Tabla 19.1, fue posible construir cuatro rutas empíricas distintas.

Los datos acerca del punto de partida y los destinos visitados en cada escenario han sido usados para presentar los resultados de los algoritmos matemáticos en la Tabla 19.2.

Tabla 19.2

Resultados de la aplicación de algoritmos matemáticos

Comparación de Resultados : Escenario 1																			
Ruta empírica:	Ruta:	PT13	PC27	PC18	PC22	PC25	PC21	PC24	PC17	PP11	PP9	PC16	PC20	PC26	PC15	PC19	PP10	PC23	PT13
	Distancia:	24189	metros																
A. Clarke y Wright	Ruta:	PT13	PC27	PC15	PC18	PP9	PC17	PC16	PC23	PC22	PC20	PC21	PC19	PP10	PC25	PC26	PC24	PP11	PT13
	Distancia:	6701	metros																
A. Vecino más Cercano	Ruta:	PT13	PP10	PP11	PC19	PC21	PC20	PC22	PC23	PC16	PC17	PP9	PC18	PC15	PC27	PC25	PC26	PC24	PT13
	Distancia:	7281	metros																
A. Búsqueda Tabú	Ruta:	PT13	PP11	PC24	PC26	PC25	PP10	PC21	PC22	PC16	PP9	PC18	PC27	PC15	PC17	PC23	PC20	PC19	PT13
	Distancia:	6430	metros																
A. Genético	Ruta:	PT13	PC19	PC21	PC23	PC15	PC27	PC18	PP9	PC16	PC17	PC22	PC20	PP10	PP11	PC24	PC26	PC25	PT13
	Distancia:	6665	metros																
Comparación de Resultados : Escenario 2																			
Ruta empírica:	Ruta:	PT12	PP10	PC19	PP9	PP11	PC16	PC20	PC17	PC15	PC18	PC24	PC26	PC21	PC23	PT12			
	Distancia:	15862	metros																
A. Clarke y Wright	Ruta:	PT12	PC18	PC15	PP9	PC17	PC16	PC23	PC20	PC21	PC19	PP10	PP11	PC26	PC24	PT12			
	Distancia:	6910	metros																
A. Vecino más Cercano	Ruta:	PT12	PC24	PC26	PP11	PP10	PC19	PC21	PC20	PC23	PC16	PC17	PP9	PC18	PC15	PT12			
	Distancia:	6904	metros																
A. Búsqueda Tabú	Ruta:	PT12	PC24	PP11	PC19	PC20	PC23	PC17	PC15	PC18	PP9	PC16	PC21	PP10	PC26	PT12			
	Distancia:	6722	metros																
A. Genético	Ruta:	PT12	PC24	PP11	PC21	PP9	PC18	PC15	PC17	PC16	PC23	PC20	PC19	PP10	PC26	PT12			
	Distancia:	6789	metros																

Comparación de Resultados : Escenario 3													
Ruta empírica:	Ruta:	PT10	PP11	PC15	PC26	PP10	PC16	PC17	PC19	PC25	PC27	PC21	PT10
	Distancia:	18083	metros										
A. Clarke y Wright	Ruta:	PT10	PC26	PC25	PP11	PP10	PC19	PC21	PC16	PC17	PC15	PC27	PT10
	Distancia:	6515	metros										
A. Vecino más Cercano	Ruta:	PT10	PC27	PC15	PC17	PC16	PC21	PC19	PP10	PP11	PC25	PC26	PT10
	Distancia:	6515	metros										
A. Búsqueda Tabú	Ruta:	PT10	PC27	PC16	PC21	PP10	PC25	PC26	PP11	PC19	PC17	PC15	PT10
	Distancia:	6442	metros										
A. Genético	Ruta:	PT10	PC15	PC17	PC19	PP11	PC26	PC25	PP10	PC21	PC16	PC27	PT10
	Distancia:	6442	metros										
Comparación de Resultados : Escenario 4													
Ruta empírica:	Ruta:	PT14	PC21	PC15	PP11	PC22	PP10	PC23	PT14				
	Distancia:	9134	metros										
A. Clarke y Wright	Ruta:	PT14	PC23	PC15	PC22	PC21	PP10	PP11	PT14				
	Distancia:	4676	metros										
A. Vecino más Cercano	Ruta:	PT14	PP11	PP10	PC21	PC22	PC23	PC15	PT14				
	Distancia:	4743	metros										
A. Búsqueda Tabú	Ruta:	PT14	PP11	PC23	PC15	PC22	PC21	PP10	PT14				
	Distancia:	4673	metros										
A. Genético	Ruta:	PT14	PP10	PC21	PC22	PC15	PC23	PP11	PT14				
	Distancia:	4673	metros										

Los resultados expuestos en la Tabla 19.2 señalan en primera impresión, que las rutas ofrecidas por los algoritmos matemáticos son mejores. La comparación porcentual se ve en Tabla 19.3.

Tabla 19.3

Comparación porcentual de los resultados

	Escenario 1		Escenario 2		Escenario 3		Escenario 4	
	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual
Ruta empírica	24189	---	15862	---	18083	---	9134	---
A. Clarke y Wright	6701	72.3%	6910	56.4%	6515	64%	4676	48.8%
A. Vecino más Cercano	7281	69.9%	6904	56.5%	6515	64%	4743	48.1%
A. Búsqueda Tabú	6430	73.4%	6722	57.6%	6442	64.4%	4673	48.8%
A. Genético	6665	72.4%	6789	57.2%	6442	64.4%	4673	48.8%

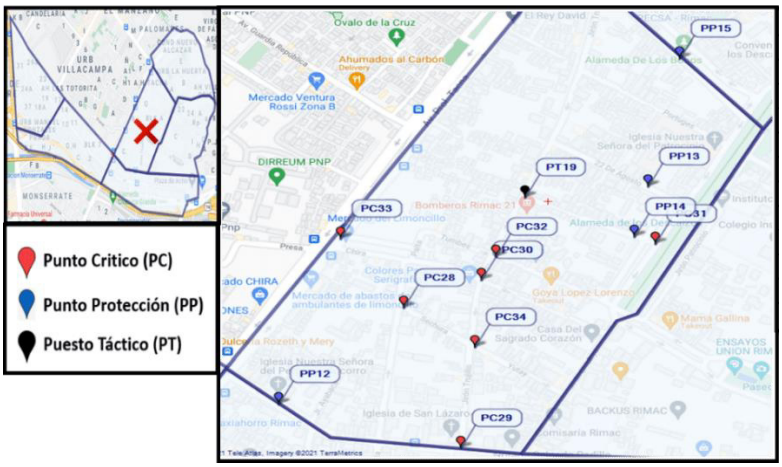
Los tres primeros escenarios indican visitar más de diez puntos en una sola ruta, estas decisiones son poco elegidas por ser muy extensas en recorrido, los algoritmos logran reducir en promedio 72%, 57% y 64% las distancias de recorrido de los vehículos para cada ruta empírica en el primer, segundo y tercer escenario respectivamente.

En el último escenario, la decisión de visitar seis puntos en un solo recorrido es común en el patrullaje, los algoritmos logran reducir en promedio el 49% la distancia del recorrido.

Análisis para el sector S4

Figura 19

Mapa del sector S4 de CPNP Rímac



Nota. La leyenda se encuentra en Anexo 9.2.

En este sector es común que sea vigilado junto al sector S5 por un solo vehículo en el primer turno de patrullaje (07:00 hasta 12:00), sin embargo, se plantea la situación ideal en que se le asignó un vehículo. Los puntos del sector estan en la Tabla 20.

Tabla 20

Información del sector S4 de CPNP Rímac

		Puntos del sector						
Origen	Táctico	PT19						
	Índice	1						
Destinos	Crítico	PC28	PC29	PC30	PC31	PC32	PC33	PC34
	Índice	1	2	3	4	5	6	7
	Protección	PP12	PP13	PP14	PP15			
	Índice	8	9	10	11			

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

Se ha construido rutas empíricas mediante la selección aleatoria de puntos. Ver la Tabla 20.1.

Tabla 20.1

Datos aleatorios para construir rutas empíricas en el S4 de CPNP Rímac

Escenario 1 : 100% (total Puntos) : 11 puntos												
O	Índice:	1										
	Punto Inicial:	PT19										
D	Aleatorios:	0.7143	0.099	0.836	0.318	0.188	0.524	0.795	0.232	0.117	0.06	0.1558
	Índice:	3	10	1	5	7	4	2	6	9	11	8
	Destino:	PC30	PP14	PC28	PC32	PC34	PC31	PC29	PC33	PP13	PP15	PP12
Escenario 2 : 80% (total Puntos) : 9 puntos												
O	Índice:	1										
	Punto Inicial:	PT19										
D	Aleatorios:	0.3399	0.234	0.178	0.902	0.566	0.927	0.693	0.422	0.351	0.789	0.545
	Índice:		10	11	2	5	1	4	7	8		
	Destino:	PP13	PP14	PP15	PC29	PC32	PC28	PC31	PC34	PP12		
Escenario 3 : 60% (total Puntos) : 7 puntos												
O	Índice:	1										
	Punto Inicial:	PT19										
D	Aleatorios:	0.5075	0.234	0.316	0.369	0.526	0.69	0.984	0.203	0.872	0.28	0.29
	Índice:		10	7	6	4	3	1				
	Destino:	PC32	PP14	PC34	PC33	PC31	PC30	PC28				
Escenario 4 : 40% (total Puntos) : 4 puntos												
O	Índice:	1										
	Punto Inicial:	PT19										
D	Aleatorios:	0.1619	0.96	0.186	0.914	0.417	0.748	0.538	0.175	0.128	0.292	0.089
	Índice:		1	7	2							
	Destino:	PP13	PC28	PC34	PC29							

Nota. Se muestra los resultados aleatorios de las fórmulas de Excel. Se denota que O es origen y D es destinos.

Los datos mostrados en la Tabla 20.1 indican que se han construido cuatro rutas empíricas. En el primer, segundo, tercer y cuarto escenario se visitan once, nueve, siete y cuatro puntos respectivamente.

Se usaron los datos acerca del punto de partida y los destinos visitados en cada escenario para presentar los resultados de los algoritmos matemáticos en la Tabla 20.2.

Tabla 20.2

Resultados de la aplicación de algoritmos matemáticos

Comparación de Resultados : Escenario 1														
Ruta empírica	Ruta:	PT19	PC30	PP14	PC28	PC32	PC34	PC31	PC29	PC33	PP13	PP15	PP12	PT19
	Distancia:	9720	metros											
A. Clarke y Wright	Ruta:	PT19	PP12	PC33	PC28	PC29	PC34	PC30	PC32	PC31	PP13	PP15	PP14	PT19
	Distancia:	4314	metros											
A. Vecino más Cercano	Ruta:	PT19	PC32	PC30	PC34	PC29	PC28	PC33	PP12	PP14	PP13	PC31	PP15	PT19
	Distancia:	4273	metros											
A. Búsqueda Tabú	Ruta:	PT19	PC32	PC30	PC28	PC33	PP12	PC29	PC34	PP14	PC31	PP13	PP15	PT19
	Distancia:	4125	metros											
A. Genético	Ruta:	PT19	PC32	PC30	PC28	PC33	PP12	PC29	PC34	PP13	PP15	PC31	PP14	PT19
	Distancia:	4154	metros											
Comparación de Resultados : Escenario 2														
Ruta empírica	Ruta:	PT19	PP13	PP14	PP15	PC29	PC32	PC28	PC31	PC34	PP12	PT19		
	Distancia:	6919	metros											
A. Clarke y Wright	Ruta:	PT19	PP12	PC28	PC29	PC34	PC32	PC31	PP13	PP15	PP14	PT19		
	Distancia:	4231	metros											
A. Vecino más Cercano	Ruta:	PT19	PC32	PC34	PC29	PC28	PP12	PP14	PP13	PC31	PP15	PT19		
	Distancia:	4190	metros											
A. Búsqueda Tabú	Ruta:	PT19	PC32	PC28	PP12	PC29	PC34	PP14	PC31	PP13	PP15	PT19		
	Distancia:	4052	metros											
A. Genético	Ruta:	PT19	PC32	PC28	PP12	PC29	PC34	PP14	PC31	PP13	PP15	PT19		
	Distancia:	4052	metros											

Comparación de Resultados : Escenario 3		
Ruta empírica	Ruta:	PT19 PC32 PP14 PC34 PC33 PC31 PC30 PC28 PT19
	Distancia:	5392 metros
A. Clarke y Wright	Ruta:	PT19 PC33 PC28 PC34 PC30 PC32 PC31 PP14 PT19
	Distancia:	3002 metros
A. Vecino más Cercano	Ruta:	PT19 PC32 PC30 PC34 PC28 PC33 PP14 PC31 PT19
	Distancia:	3019 metros
A. Búsqueda Tabú	Ruta:	PT19 PC32 PC30 PC33 PC28 PC34 PC31 PP14 PT19
	Distancia:	2971 metros
A. Genético	Ruta:	PT19 PP14 PC31 PC34 PC28 PC33 PC30 PC32 PT19
	Distancia:	2971 metros
Comparación de Resultados : Escenario 4		
Ruta empírica	Ruta:	PT19 PP13 PC28 PC34 PC29 PT19
	Distancia:	2629 metros
A. Clarke y Wright	Ruta:	PT19 PC29 PC28 PC34 PP13 PT19
	Distancia:	2488 metros
A. Vecino más Cercano	Ruta:	PT19 PC34 PC29 PC28 PP13 PT19
	Distancia:	2478 metros
A. Búsqueda Tabú	Ruta:	PT19 PC28 PC29 PC34 PP13 PT19
	Distancia:	2442 metros
A. Genético	Ruta:	PT19 PP13 PC34 PC29 PC28 PT19
	Distancia:	2442 metros

En la Tabla 20.2 se muestra que las rutas diseñadas por los algoritmos son mejores que las rutas decididas empíricamente. Ver en la Tabla 20.3.

Tabla 20.3

Comparación porcentual de los resultados

	Escenario 1		Escenario 2		Escenario 3		Escenario 4	
	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual
Ruta empírica	9720	---	6919	---	5392	---	2629	---
A. Clarke y Wright	4314	55.6%	4231	38.8%	3002	44.3%	2488	5%
A. Vecino más Cercano	4273	56%	4190	39.4%	3019	44%	2478	5.7%
A. Búsqueda Tabú	4125	57.6%	4052	41.4%	2971	44.9%	2442	7%
A. Genético	4154	57.3%	4052	41.4%	2971	44.9%	2442	7%

El primer escenario plantea la situación en que se decida visitar más de diez puntos en un solo recorrido, si bien es cierto esta decisión no es muy frecuente, con el aporte de algoritmos se pudo reducir en promedio un 57% la distancia del recorrido de la ruta empírica.

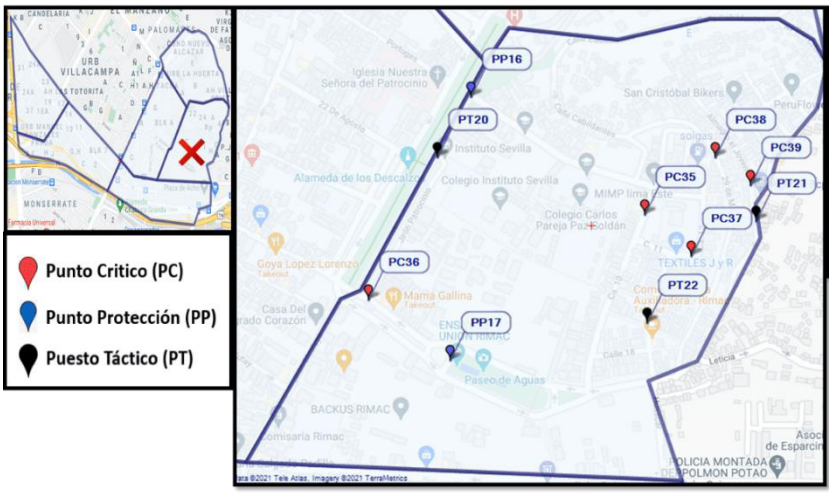
Los dos siguientes escenarios, son decisiones generalmente cotidianas porque los recorridos consisten en visitar de siete a nueve puntos, con el uso de los algoritmos matemáticos se logró reducir en promedio 40% y 45% en el segundo y tercer escenario respectivamente las distancias de recorrido de los vehículos para cada ruta empírica.

Con respecto al cuarto escenario, esta decisión no se daría porque son recorridos cortos, aun así, los algoritmos logran reducir en promedio un 6% la distancia de recorrido de los vehículos en la ruta empírica de este escenario. No se aprecia una reducción considerable pero aun así se ve una mejora.

Análisis para el sector S5

Figura 20

Mapa del sector S5 de CPNP Rímac



Nota. La leyenda se encuentra en Anexo 9.2.

Como se mencionó en el caso del sector S4, en el primer turno de patrullaje móvil es asignado un vehículo a custodiar los sectores S4 y S5, sin embargo, se plantea la situación ideal en que se le asignó un vehículo en el sector S5.

La Tabla 21 indica que hay siete puntos a visitar y tres puestos tácticos para el inicio del recorrido.

Tabla 21

Información del sector S5 de CPNP Rímac

		Puntos del sector				
Origen	Táctico	PT20	PT21	PT22		
	Índice	1	2	3		
Destinos	Crítico	PC35	PC36	PC37	PC38	PC39
	Índice	1	2	3	4	5
	Protección	PP16	PP17			
	Índice	6	7			

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

Con los datos de la Tabla 21 se han construido rutas empíricas mediante la selección aleatoria de puntos. Ver la Tabla 21.1.

Tabla 21.1

Datos aleatorios para construir rutas empíricas en el S5 de CPNP Rímac

Escenario 1 : 100% (total Puntos) : 7 puntos								
O	Índice:	2						
	Punto Inicial:	PT21						
D	Aleatorios:	0.434	0.469	0.825	0.563	0.65	0.68	0.196
	Índice:	6	5	1	4	3	2	7
	Destino:	PP16	PC39	PC35	PC38	PC37	PC36	PP17
Escenario 2 : 80% (total Puntos) : 6 puntos								
O	Índice:	3						
	Punto Inicial:	PT22						
D	Aleatorios:	0.533	0.22	0.333	0.285	0.575	0.702	0.1
	Índice:	3	6	4	5	2	1	
	Destino:	PC37	PP16	PC38	PC39	PC36	PC35	
Escenario 3 : 60% (total Puntos) : 4 puntos								
O	Índice:	1						
	Punto Inicial:	PT20						
D	Aleatorios:	0.371	0.433	0.145	0.115	0.669	0.677	0.085
	Índice:	4	3	5	6			
	Destino:	PC38	PC37	PC39	PP16			

Nota. Se muestra los resultados aleatorios de las fórmulas de Excel. Se denota que O es origen y D es destinos.

Se indica en la Tabla 21.1 que han sido construidas tres rutas empíricas. En el primer, segundo y tercer escenario se visitan siete, seis y cuatro puntos respectivamente.

No se planteó un cuarto escenario porque tendría como resultado la construcción de un recorrido que involucre visitar menos de cuatro destinos, esta decisión no es tomada en cuenta en los patrullajes.

Se presentan los resultados de los algoritmos matemáticos en la Tabla 21.2.

Tabla 21.2

Resultados de la aplicación de algoritmos matemáticos

Comparación de Resultados : Escenario 1										
Ruta empírica	Ruta:	PT21	PP16	PC39	PC35	PC38	PC37	PC36	PP17	PT21
	Distancia:	6971	metros							
A. Clarke y Wright	Ruta:	PT21	PP17	PC36	PP16	PC35	PC37	PC38	PC39	PT21
	Distancia:	3762	metros							
A. Vecino más Cercano	Ruta:	PT21	PC39	PC38	PC37	PC35	PP16	PP17	PC36	PT21
	Distancia:	3904	metros							
A. Búsqueda Tabú	Ruta:	PT21	PC39	PC38	PP16	PC36	PP17	PC35	PC37	PT21
	Distancia:	3721	metros							
A. Genético	Ruta:	PT21	PC39	PC38	PP16	PC36	PP17	PC35	PC37	PT21
	Distancia:	3721	metros							
Comparación de Resultados : Escenario 2										
Ruta empírica	Ruta:	PT22	PC37	PP16	PC38	PC39	PC36	PC35	PT22	
	Distancia:	5782	metros							
A. Clarke y Wright	Ruta:	PT22	PP16	PC36	PC35	PC38	PC39	PC37	PT22	
	Distancia:	3687	metros							
A. Vecino más Cercano	Ruta:	PT22	PC35	PC37	PC38	PC39	PP16	PC36	PT22	
	Distancia:	3796	metros							
A. Búsqueda Tabú	Ruta:	PT22	PC37	PC39	PC38	PC35	PP16	PC36	PT22	
	Distancia:	3661	metros							
A. Genético	Ruta:	PT22	PC36	PP16	PC35	PC38	PC39	PC37	PT22	
	Distancia:	3661	metros							
Comparación de Resultados : Escenario 3										
Ruta empírica	Ruta:	PT20	PC38	PC37	PC39	PP16	PT20			
	Distancia:	3215	metros							
A. Clarke y Wright	Ruta:	PT20	PC39	PC38	PC37	PP16	PT20			
	Distancia:	2996	metros							
A. Vecino más Cercano	Ruta:	PT20	PP16	PC37	PC38	PC39	PT20			
	Distancia:	2996	metros							

Comparación de Resultados : Escenario 3							
A. Búsqueda Tabú	Ruta:	PT20	PP16	PC38	PC39	PC37	PT20
	Distancia:	2959	metros				
A. Genético	Ruta:	PT20	PP16	PC38	PC39	PC37	PT20
	Distancia:	2959	metros				

En la Tabla 21.2, las rutas diseñadas por los algoritmos matemáticos son mejores que las decididas empíricamente, se ve en las distancias de recorrido. La comparación porcentual se ve en Tabla 21.3.

Tabla 21.3

Comparación porcentual de los resultados

	Escenario 1		Escenario 2		Escenario 3	
	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual
Ruta empírica	6971	---	5782	---	3215	---
A. Clarke y Wright	3762	46%	3687	36.2%	2996	6.8%
A. Vecino más Cercano	3904	44%	3796	34.3%	2996	6.8%
A. Búsqueda Tabú	3721	46.6%	3661	36.7%	2959	8%
A. Genético	3721	46.6%	3661	36.7%	2959	8%

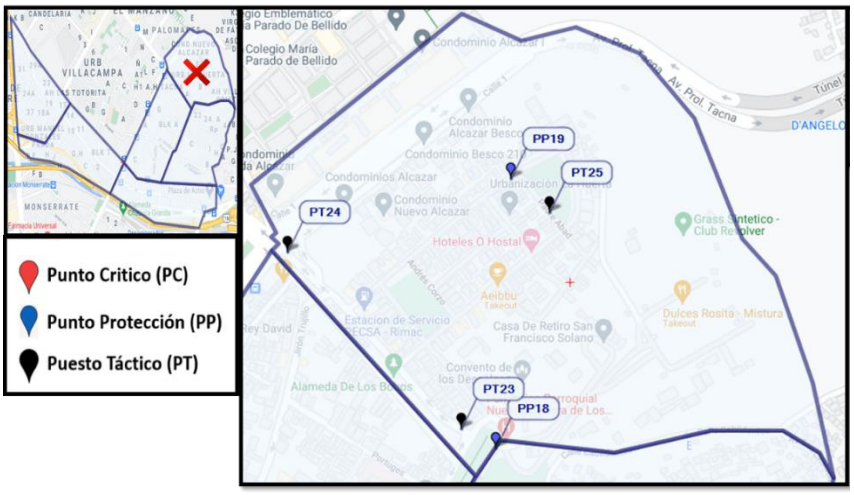
En los dos primeros escenarios, la decisión es realizar un recorrido para visitar de cinco a nueve puntos, en esta situación con el aporte de los algoritmos se reduce en promedio un 46% y 36% en el primer y segundo escenario las distancias de recorrido.

El tercer escenario indica visitar cuatro destinos, esta decisión no sería la mejor porque son recorridos cortos, los algoritmos logran reducir en promedio en 8% la distancia de las ruta empírica, no se evidencia una gran reducción pero se ve una mejora.

Análisis para el sector S6

Figura 21

Mapa del sector S6 de CPNP Rímac



Nota. La leyenda se encuentra en Anexo 9.2.

Este sector no tiene puntos críticos a custodiar sino puntos de protección, porque es una zona cercada por seguridad particular. La Tabla 22 muestra la información del sector.

Tabla 22

Información del sector S6 de CPNP Rímac

Puntos del sector				
Origen	Táctico	PT23	PT24	PT25
	Índice	1	2	3
Destinos	Protección	PP18	PP19	
	Índice	1	2	

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

En la Tabla 23 se compara las distancias de las rutas cuando se toma una decisión empírica y cuando se toma una decisión en base a técnicas matemáticas.

Tabla 23

Resultados finales para la CPNP Rímac

Sectores	Escenarios	Distancia de la ruta empírica (metros)	Promedio de distancias de rutas con Heurísticas y Metaheurísticas (metros)	Estado
S1	Escenario 1	7274	3081	Reducción
	Escenario 2	5646	3021.5	Reducción
	Escenario 3	4293	2640.75	Reducción
	Escenario 4	4367	2714.5	Reducción
S2	Escenario 1	6343	4759.5	Reducción
	Escenario 2	7503	4682	Reducción
	Escenario 3	9152	4950.25	Reducción
	Escenario 4	4925	4768	Reducción
S3	Escenario 1	24189	6769.25	Reducción
	Escenario 2	15862	6831.25	Reducción
	Escenario 3	18083	6478.5	Reducción
	Escenario 4	9134	4691.25	Reducción
S4	Escenario 1	9720	4216.5	Reducción
	Escenario 2	6919	4131.25	Reducción
	Escenario 3	5392	2990.75	Reducción
	Escenario 4	2629	2462.5	Reducción
S5	Escenario 1	6971	3777	Reducción
	Escenario 2	5782	3701.25	Reducción
	Escenario 3	3215	2977.5	Reducción
S6	No se realiza patrullaje móvil			

Nota. La ultima columna puntualiza el efecto de usar técnicas matemáticas sobre decisiones empíricas.

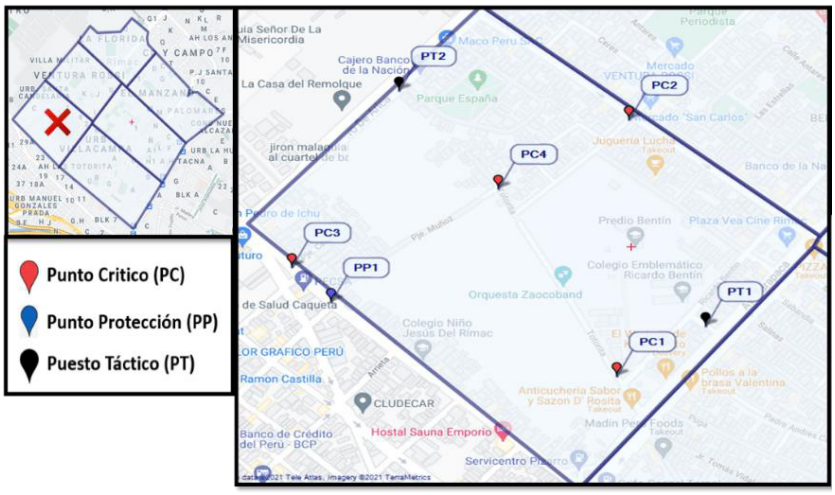
Según la Tabla 23 se puede observar que la aplicación de heurísticas y metaheurísticas para diseñar rutas, han logrado reducir la distancia del recorrido de los vehículos en las rutinas de patrullaje móvil para la comisaría PNP Rímac.

5.4.3. Análisis de resultados para CPNP Manzano

Análisis para el sector S1

Figura 22

Mapa del sector S1 de CPNP Manzano



Nota. La leyenda se encuentra en Anexo 9.3.

En este sector los vehículos se movilizan en los alrededores (avenidas principales), esto se debe a que gran parte del sector es una urbanización cerrada y sin mucho acceso al tránsito vehicular. Se ve reflejado en la Figura 22 que la mayoría de los puntos se encuentran en las proximidades de esta área. En esta medida la actividad principal es el patrullaje táctico en cualquier punto. La Tabla 24 muestra la información del sector.

Tabla 24

Información del sector S1 de CPNP Manzano

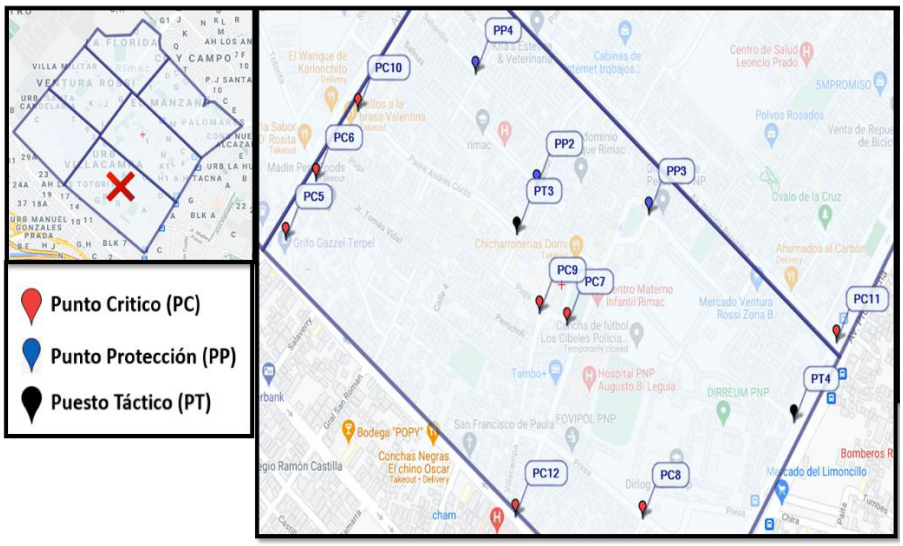
Puntos del sector					
Origen	Táctico Índice	PT1 1	PT2 2		
Destinos	Crítico Índice	PC1 1	PC2 2	PC3 3	PC4 4
	Protección Índice	PP1 5			

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

Análisis para el sector S2

Figura 23

Mapa del sector S2 de CPNP Manzano



Nota. La leyenda se encuentra en Anexo 9.3.

En la Tabla 25, se ve que en este sector existen once puntos que son vigilados en las rutinas del patrullaje y dos puestos tácticos que son considerados como puntos de partida del patrullaje móvil.

Tabla 25

Información del sector S2 de CPNP Manzano

Puntos del sector					
Origen	Táctico	PT3	PT4		
	Índice	1	2		
Destinos	Crítico	PC5	PC6	PC7	PC8
	Índice	1	2	3	4
	Crítico	PC9	PC10	PC11	PC12
	Índice	5	6	7	8
	Protección	PP2	PP3	PP4	
	Índice	9	10	11	

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

De los datos mostrados en la Tabla 25 se han seleccionado aleatoriamente puntos para construir rutas empíricas. La selección está en la Tabla 25.1.

Tabla 25.1

Datos aleatorios para construir rutas empíricas en el S2 de CPNP Manzano

Escenario 1 : 100% (total Puntos) : 11 puntos												
O	Índice:	1										
	Punto Inicial:	PT3										
D	Aleatorios:	0.516	0.59	0.382	0.268	0.845	0.388	0.431	0.955	0.821	0.765	0.137
	Índice:	6	5	9	10	2	8	7	1	3	4	11
	Destino:	PC10	PC9	PP2	PP3	PC6	PC12	PC11	PC5	PC7	PC8	PP4
Escenario 2 : 80% (total Puntos) : 9 puntos												
O	Índice:	2										
	Punto Inicial:	PT4										
D	Aleatorios:	0.343	0.852	0.888	0.039	0.513	0.17	0.519	0.651	0.757	0.142	0.37
	Índice:	8	2	1	11	6	9	5	4	3		
	Destino:	PC12	PC6	PC5	PP4	PC10	PP2	PC9	PC8	PC7		
Escenario 3 : 60% (total Puntos) : 7 puntos												
O	Índice:	2										
	Punto Inicial:	PT4										
D	Aleatorios:	0.687	0.261	0.361	0.952	0.201	0.164	0.555	0.925	0.647	0.895	0.108
	Índice:	4	8	7	1	9	10	6				
	Destino:	PC8	PC12	PC11	PC5	PP2	PP3	PC10				
Escenario 4 : 40% (total Puntos) : 4 puntos												
O	Índice:	1										
	Punto Inicial:	PT3										
D	Aleatorios:	0.813	0.615	0.76	0.202	0.665	0.721	0.902	0.219	0.555	0.912	0.16
	Índice:	3	7	4	10							
	Destino:	PC7	PC11	PC8	PP3							

Nota. Se muestra los resultados aleatorios de las fórmulas de Excel. Se denota que O es origen y D es destinos

Fue posible construir cuatro rutas empíricas distintas según lo mostrado en la Tabla 25.1.

Los datos seleccionados en cada escenario han sido utilizados para presentar los resultados de los algoritmos.

Tabla 25.2

Resultados de la aplicación de algoritmos matemáticos

Comparación de Resultados : Escenario 1													
Ruta empírica	Ruta:	PT3	PC10	PC9	PP2	PP3	PC6	PC12	PC11	PC5	PC7	PC8	PP4	PT3
	Distancia:	13548	metros											
A. Clarke y Wright	Ruta:	PT3	PC6	PC5	PC10	PP4	PC12	PC8	PC11	PP3	PC7	PC9	PP2	PT3
	Distancia:	6408	metros											
A. Vecino más Cercano	Ruta:	PT3	PP2	PC9	PC7	PC12	PP4	PC10	PC6	PC5	PC8	PP3	PC11	PT3
	Distancia:	7241	metros											
A. Búsqueda Tabú	Ruta:	PT3	PP2	PC9	PC7	PP3	PC11	PC8	PC12	PC5	PC6	PC10	PP4	PT3
	Distancia:	6228	metros											
A. Genético	Ruta:	PT3	PP4	PC10	PC6	PC5	PC12	PC9	PC7	PC8	PC11	PP3	PP2	PT3
	Distancia:	6248	metros											
Comparación de Resultados : Escenario 2													
Ruta empírica	Ruta:	PT4	PC12	PC6	PC5	PP4	PC10	PP2	PC9	PC8	PC7	PT4		
	Distancia:	7878	metros											
A. Clarke y Wright	Ruta:	PT4	PC6	PC5	PC10	PP4	PP2	PC9	PC12	PC7	PC8	PT4		
	Distancia:	6138	metros											
A. Vecino más Cercano	Ruta:	PT4	PC8	PC7	PC9	PP2	PC12	PP4	PC10	PC6	PC5	PT4		
	Distancia:	6150	metros											
A. Búsqueda Tabú	Ruta:	PT4	PC8	PC12	PC5	PC6	PC10	PP4	PP2	PC9	PC7	PT4		
	Distancia:	5710	metros											
A. Genético	Ruta:	PT4	PC12	PC5	PC6	PC10	PP4	PP2	PC9	PC7	PC8	PT4		
	Distancia:	5745	metros											
Comparación de Resultados : Escenario 3													
Ruta empírica	Ruta:	PT4	PC8	PC12	PC11	PC5	PP2	PP3	PC10	PT4				
	Distancia:	12242	metros											
A. Clarke y Wright	Ruta:	PT4	PC10	PC5	PC12	PP2	PP3	PC8	PC11	PT4				
	Distancia:	6601	metros											
A. Vecino más Cercano	Ruta:	PT4	PC11	PP3	PC8	PP2	PC12	PC10	PC5	PT4				
	Distancia:	6642	metros											

Comparación de Resultados : Escenario 3										
A. Búsqueda Tabú	Ruta:	PT4	PC11	PP3	PP2	PC10	PC5	PC12	PC8	PT4
	Distancia:	6070	metros							
A. Genético	Ruta:	PT4	PC8	PC12	PC5	PC10	PP2	PP3	PC11	PT4
	Distancia:	6070	metros							
Comparación de Resultados : Escenario 4										
Ruta empírica	Ruta:	PT3	PC7	PC11	PC8	PP3	PT3			
	Distancia:	3821	metros							
A. Clarke y Wright	Ruta:	PT3	PP3	PC11	PC8	PC7	PT3			
	Distancia:	3553	metros							
A. Vecino más Cercano	Ruta:	PT3	PC7	PC8	PP3	PC11	PT3			
	Distancia:	3789	metros							
A. Búsqueda Tabú	Ruta:	PT3	PC7	PC8	PC11	PP3	PT3			
	Distancia:	3553	metros							
A. Genético	Ruta:	PT3	PC7	PC8	PC11	PP3	PT3			
	Distancia:	3553	metros							

Como se ve en la Tabla 25.2, los diseños de rutas con los algoritmos matemáticos son mejores.

Tabla 25.3

Comparación porcentual de los resultados

	Escenario 1		Escenario 2		Escenario 3		Escenario 4	
	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual
Ruta empírica	13548	---	7878	---	12242	---	3821	---
A. Clarke y Wright	6408	52.7%	6138	22.1%	6601	46.1%	3553	7%
A. Vecino más Cercano	7241	46.6%	6150	21.9%	6642	45.7%	3789	1%
A. Búsqueda Tabú	6228	54%	5710	27.5%	6070	50.4%	3553	7%
A. Genético	6248	53.9%	5745	27.1%	6070	50.4%	3553	7%

El primer escenario es visitar once puntos, esta decisión no es muy frecuente por ser considerado muy extenso, utilizando algoritmos matemáticos se logró reducir en promedio 52% la distancia de recorrido de los vehículos.

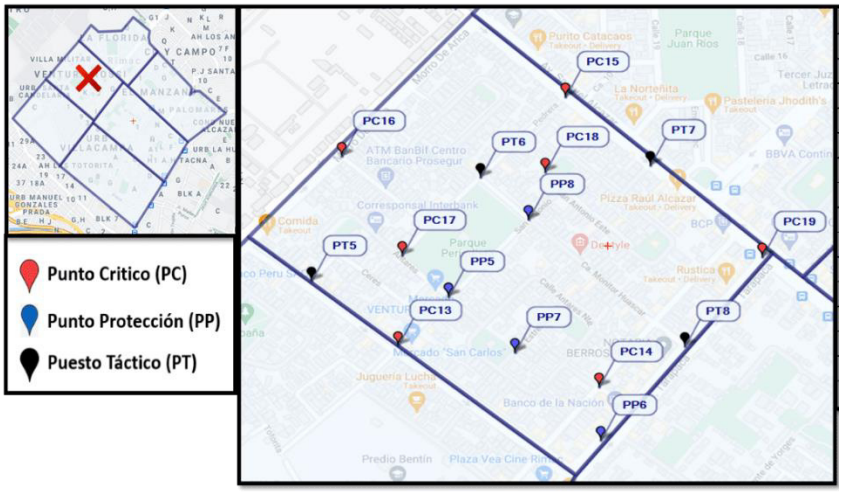
El segundo y tercer escenario consiste en realizar trayectorias que involucre visitar de cinco a nueve puntos, usando algoritmos se reduce en promedio 25% y 48% las distancias de las rutas construidas empíricamente.

Para el último escenario, es muy poco probable la decisión de realizar una trayectoria que consista en cuatro puntos, esto porque son recorridos muy cortos, los algoritmos logran reducir en promedio 5% la distancia de recorrido de la ruta empírica.

Análisis para el sector S3

Figura 24

Mapa del sector S3 de CPNP Manzano



Nota. La leyenda se encuentra en Anexo 9.3.

La información de este sector está en la Tabla 26.

Tabla 26

Información del sector S3 de CPNP Manzano

		Puntos del sector			
Origen	Táctico	PT5	PT6	PT7	PT8
	Índice	1	2	3	4
Destinos	Crítico	PC13	PC14	PC15	PC16
	Índice	1	2	3	4
	Crítico	PC17	PC18	PC19	
	Índice	5	6	7	
	Protección	PP5	PP6	PP7	PP8
	Índice	8	9	10	11

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

Se seleccionó aleatoriamente puntos para construir rutas empíricas.

Tabla 26.1

Datos aleatorios para construir rutas empíricas en el S3 de CPNP Manzano

Escenario 1 : 100% (total Puntos) : 11 puntos												
O	Índice:	2										
	Punto Inicial:	PT6										
D	Aleatorios:	0.761	0.662	0.153	0.591	0.266	0.07	0.54	0.081	0.679	0.169	0.439
	Índice:	1	3	9	4	7	11	5	10	2	8	6
	Destino:	PC13	PC15	PP6	PC16	PC19	PP8	PC17	PP7	PC14	PP5	PC18
Escenario 2 : 80% (total Puntos) : 9 puntos												
O	Índice:	1										
	Punto Inicial:	PT5										
D	Aleatorios:	0.569	0.761	0.564	0.091	0.775	0.357	0.132	0.97	0.519	0.926	0.733
	Índice:	6	4	7	11	3	9	10	1	8		
	Destino:	PC18	PC16	PC19	PP8	PC15	PP6	PP7	PC13	PP5		
Escenario 3 : 60% (total Puntos) : 7 puntos												
O	Índice:	4										
	Punto Inicial:	PT8										
D	Aleatorios:	0.461	0.1	0.307	0.966	0.24	0.439	0.449	0.877	0.89	0.4	0.271
	Índice:	4	11	8	1	10	6	5				
	Destino:	PC16	PP8	PP5	PC13	PP7	PC18	PC17				
Escenario 4 : 40% (total Puntos) : 4 puntos												
O	Índice:	3										
	Punto Inicial:	PT7										
D	Aleatorios:	0.128	0.094	0.437	0.958	0.747	0.777	0.47	0.883	0.353	0.908	0.562
	Índice:	10	11	8	1							
	Destino:	PP7	PP8	PP5	PC13							

Nota. Se muestra los resultados aleatorios de las fórmulas de Excel. Se denota que O es origen y D es destinos.

El número de puntos a visitar por escenario es idéntico al del sector anterior. Los resultados en la Tabla 26.2.

Tabla 26.2

Resultados de la aplicación de algoritmos matemáticos

Comparación de Resultados : Escenario 1														
Ruta empírica	Ruta:	PT6	PC13	PC15	PP6	PC16	PC19	PP7	PC17	PP8	PC14	PP5	PC18	PT6
	Distancia:	8657	metros											
A. Clarke y Wright	Ruta:	PT6	PP6	PC14	PC19	PC15	PC18	PP8	PP7	PP5	PC13	PC16	PC17	PT6
	Distancia:	4185	metros											
A. Vecino más Cercano	Ruta:	PT6	PP5	PC17	PC13	PC16	PP7	PP8	PC18	PC15	PC14	PP6	PC19	PT6
	Distancia:	4426	metros											
A. Búsqueda Tabú	Ruta:	PT6	PC17	PC16	PC13	PP5	PP7	PC14	PP6	PC19	PC15	PC18	PP8	PT6
	Distancia:	4021	metros											
A. Genético	Ruta:	PT6	PP8	PC18	PC15	PC19	PP6	PC14	PP7	PP5	PC13	PC16	PC17	PT6
	Distancia:	4021	metros											
Comparación de Resultados : Escenario 2														
Ruta empírica	Ruta:	PT5	PC18	PC16	PC19	PP8	PC15	PP6	PP7	PC13	PP5	PT5		
	Distancia:	7255	metros											
A. Clarke y Wright	Ruta:	PT5	PP6	PC19	PC15	PC18	PP8	PP7	PP5	PC13	PC16	PT5		
	Distancia:	4362	metros											
A. Vecino más Cercano	Ruta:	PT5	PC16	PC13	PP5	PP7	PP8	PC18	PC15	PP6	PC19	PT5		
	Distancia:	4585	metros											
A. Búsqueda Tabú	Ruta:	PT5	PC16	PP8	PC18	PC15	PC19	PP6	PP7	PP5	PC13	PT5		
	Distancia:	4200	metros											
A. Genético	Ruta:	PT5	PC13	PP5	PP7	PP6	PC19	PC15	PC18	PP8	PC16	PT5		
	Distancia:	4200	metros											
Comparación de Resultados : Escenario 3														
Ruta empírica	Ruta:	PT8	PC16	PP8	PP5	PC13	PP7	PC18	PC17	PT8				
	Distancia:	5646	metros											
A. Clarke y Wright	Ruta:	PT8	PC17	PC16	PC13	PP5	PP7	PP8	PC18	PT8				
	Distancia:	3313	metros											
A. Vecino más Cercano	Ruta:	PT8	PC18	PP8	PP7	PP5	PC17	PC13	PC16	PT8				
	Distancia:	3385	metros											

Comparación de Resultados : Escenario 3									
A. Búsqueda Tabú	Ruta:	PT8	PC18	PP8	PC17	PC16	PC13	PP5	PP7	PT8
	Distancia:	3214	metros							
A. Genético	Ruta:	PT8	PP7	PP5	PC13	PC16	PC17	PP8	PC18	PT8
	Distancia:	3214	metros							
Comparación de Resultados : Escenario 4										
Ruta empírica	Ruta:	PT7	PP7	PP8	PP5	PC13	PT7			
	Distancia:	2527	metros							
A. Clarke y Wright	Ruta:	PT7	PP5	PC13	PP7	PP8	PT7			
	Distancia:	2361	metros							
A. Vecino más Cercano	Ruta:	PT7	PP8	PP7	PP5	PC13	PT7			
	Distancia:	2373	metros							
A. Búsqueda Tabú	Ruta:	PT7	PP8	PP5	PC13	PP7	PT7			
	Distancia:	2296	metros							
A. Genético	Ruta:	PT7	PP7	PC13	PP5	PP8	PT7			
	Distancia:	2296	metros							

Las rutas obtenidas por los algoritmos son mejores que las empíricas, esto se nota por las distancias de recorrido.

Tabla 26.3

Comparación porcentual de los resultados

	Escenario 1		Escenario 2		Escenario 3		Escenario 4	
	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual
Ruta empírica	8657	---	7255	---	5646	---	2527	---
A. Clarke y Wright	4185	51.7%	4362	39.9%	3313	41.3%	2361	7%
A. Vecino más Cercano	4426	48.9%	4585	36.8%	3385	40%	2373	6%
A. Búsqueda Tabú	4021	53.6%	4200	42.1%	3214	43.1%	2296	9%
A. Genético	4021	53.6%	4200	42.1%	3214	43.1%	2296	9%

El primer escenario indica visitar once puntos, esta decisión no es frecuente, pero con los algoritmos se logró reducir en promedio 52% la distancia de recorrido.

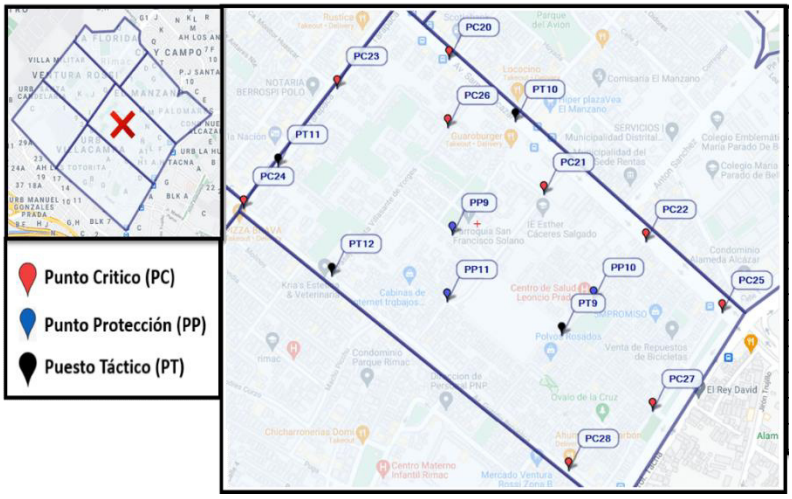
Los dos siguientes escenarios indican que los vehículos realicen rutinas de patrullaje en visitar de siete a nueve puntos, usando algoritmos se reduce en promedio 40% y 42% las distancias de las rutas empíricas construidas.

El último escenario indica visitar cuatro destinos, esta decision es pocas veces tomada en cuenta, el uso de los algoritmos matemáticos logra reducir en promedio 5% la distancia de recorrido de los vehículos en la ruta empírica de este escenario.

Análisis para el sector S4

Figura 25

Mapa del sector S4 de CPNP Manzano



Nota. La leyenda se encuentra en Anexo 9.3.

La información del sector en la Tabla 27.

Tabla 27

Información del sector S4 de CPNP Manzano

		Puntos del sector				
Origen	Táctico	PT9	PT10	PT11	PT12	
	Índice	1	2	3	4	
Destinos	Crítico	PC20	PC21	PC22	PC23	PC24
	Índice	1	2	3	4	5
	Crítico	PC25	PC26	PC27	PC28	
	Índice	6	7	8	9	
	Protección	PP9	PP10	PP11		
	Índice	10	11	12		

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

Se han construido rutas empíricas mediante la selección aleatoria de puntos.

Tabla 27.1

Datos aleatorios para construir rutas empíricas en el S4 de CPNP Manzano

Escenario 1 : 100% (total Puntos) : 12 puntos													
O	Índice:	4											
	Punto Inicial:	PT12											
D	Aleatorios:	0.993	0.604	0.107	0.12	0.034	0.065	0.283	0.222	0.441	0.738	0.781	0.843
	Índice:	1	5	10	9	12	11	7	8	6	4	3	2
	Destino:	PC20	PC24	PP9	PC28	PP11	PP10	PC26	PC27	PC25	PC23	PC22	PC21
Escenario 2 : 80% (total Puntos) : 10 puntos													
O	Índice:	2											
	Punto Inicial:	PT10											
D	Aleatorios:	0.732	0.405	0.805	0.483	0.825	0.171	0.579	0.432	0.599	0.731	0.95	0.675
	Índice:	4	11	3	9	2	12	8	10	7	5		
	Destino:	PC23	PP10	PC22	PC28	PC21	PP11	PC27	PP9	PC26	PC24		
Escenario 3 : 60% (total Puntos) : 7 puntos													
O	Índice:	3											
	Punto Inicial:	PT11											
D	Aleatorios:	0.288	0.831	0.043	0.173	0.135	0.344	0.239	0.614	0.646	0.321	0.584	0.328
	Índice:	8	1	12	10	11	5	9					
	Destino:	PC27	PC20	PP11	PP9	PP10	PC24	PC28					
Escenario 4 : 40% (total Puntos) : 5 puntos													
O	Índice:	1											
	Punto Inicial:	PT9											
D	Aleatorios:	0.955	0.838	0.853	0.771	0.337	0.893	0.043	0.243	0.53	0.026	0.235	0.62
	Índice:	1	4	3	5	8							
	Destino:	PC20	PC23	PC22	PC24	PC27							

Nota. Se muestra los resultados aleatorios de las fórmulas de Excel. Se denota que O es origen y D es destinos.

En la Tabla 27.1, se han construido cuatro rutas distintas. En el primer, segundo, tercer y cuarto escenario se visitan doce, diez, siete y cinco puntos respectivamente.

Los puntos seleccionados en cada escenario han sido usados para presentar los resultados de los algoritmos matemáticos en la Tabla 27.2.

Tabla 27.2

Resultados de la aplicación de algoritmos matemáticos

Comparación de Resultados : Escenario 1															
Ruta empírica	Ruta:	PT12	PC20	PC24	PP9	PC28	PP11	PP10	PC26	PC27	PC25	PC23	PC22	PC21	PT12
	Distancia:	13545	metros												
A. Clarke y Wright	Ruta:	PT12	PC27	PC25	PC22	PP10	PC28	PC21	PP9	PP11	PC26	PC20	PC23	PC24	PT12
	Distancia:	6187	metros												
A. Vecino más Cercano	Ruta:	PT12	PC23	PC24	PP11	PP9	PC26	PC20	PC21	PP10	PC28	PC27	PC22	PC25	PT12
	Distancia:	6706	metros												
A. Búsqueda Tabú	Ruta:	PT12	PC24	PC23	PC20	PC26	PC21	PP10	PC22	PC25	PC27	PC28	PP9	PP11	PT12
	Distancia:	5723	metros												
A. Genético	Ruta:	PT12	PC26	PC20	PC21	PP10	PC22	PC25	PC27	PC28	PP9	PP11	PC23	PC24	PT12
	Distancia:	5805	metros												
Comparación de Resultados : Escenario 2															
Ruta empírica	Ruta:	PT10	PC23	PP10	PC22	PC28	PC21	PP11	PC27	PP9	PC26	PC24	PT10		
	Distancia:	9397	metros												
A. Clarke y Wright	Ruta:	PT10	PC24	PC23	PP11	PP9	PC26	PC28	PC27	PC22	PP10	PC21	PT10		
	Distancia:	5273	metros												
A. Vecino más Cercano	Ruta:	PT10	PC21	PP10	PC28	PC27	PC22	PP9	PP11	PC26	PC23	PC24	PT10		
	Distancia:	5550	metros												
A. Búsqueda Tabú	Ruta:	PT10	PC21	PP10	PC22	PC27	PC28	PP9	PP11	PC24	PC23	PC26	PT10		
	Distancia:	5008	metros												
A. Genético	Ruta:	PT10	PC26	PC23	PC24	PP9	PP11	PC28	PC27	PC22	PP10	PC21	PT10		
	Distancia:	5037	metros												

Comparación de Resultados : Escenario 3										
Ruta empírica	Ruta:	PT11	PC27	PC20	PP11	PP9	PP10	PC24	PC28	PT11
	Distancia:	10083	metros							
A. Clarke y Wright	Ruta:	PT11	PP10	PC27	PC28	PP9	PP11	PC20	PC24	PT11
	Distancia:	5112	metros							
A. Vecino más Cercano	Ruta:	PT11	PC24	PP11	PP9	PC20	PC28	PP10	PC27	PT11
	Distancia:	5256	metros							
A. Búsqueda Tabú	Ruta:	PT11	PC24	PP11	PP9	PC28	PC27	PP10	PC20	PT11
	Distancia:	4849	metros							
A. Genético	Ruta:	PT11	PC24	PP11	PP9	PC28	PC27	PP10	PC20	PT11
	Distancia:	4849	metros							
Comparación de Resultados : Escenario 4										
Ruta empírica	Ruta:	PT9	PC20	PC23	PC22	PC24	PC27	PT9		
	Distancia:	8238	metros							
A. Clarke y Wright	Ruta:	PT9	PC24	PC23	PC20	PC22	PC27	PT9		
	Distancia:	4587	metros							
A. Vecino más Cercano	Ruta:	PT9	PC22	PC27	PC20	PC23	PC24	PT9		
	Distancia:	4917	metros							
A. Búsqueda Tabú	Ruta:	PT9	PC27	PC22	PC20	PC23	PC24	PT9		
	Distancia:	4857	metros							
A. Genético	Ruta:	PT9	PC27	PC22	PC20	PC23	PC24	PT9		
	Distancia:	4857	metros							

Nota. Se muestra los resultados aleatorios de las fórmulas de Excel.

En la Tabla 27.2 se ve que las soluciones de los algoritmos son mejores, las distancias de recorrido son menores a las trayectorias decididas empíricamente. La comparación porcentual se ve en Tabla 27.3.

Tabla 27.3

Comparación porcentual de los resultados

	Escenario 1		Escenario 2		Escenario 3		Escenario 4	
	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual
Ruta empírica	13545	---	9397	---	10083	---	8238	---
A. Clarke y Wright	6187	54.3%	5273	43.9%	5112	49.3%	4587	44%
A. Vecino más Cercano	6706	50.5%	5550	40.9%	5256	47.9%	4917	40%
A. Búsqueda Tabú	5723	57.7%	5008	46.7%	4849	51.9%	4857	41%
A. Genético	5805	57.1%	5037	46.4%	4849	51.9%	4857	41%

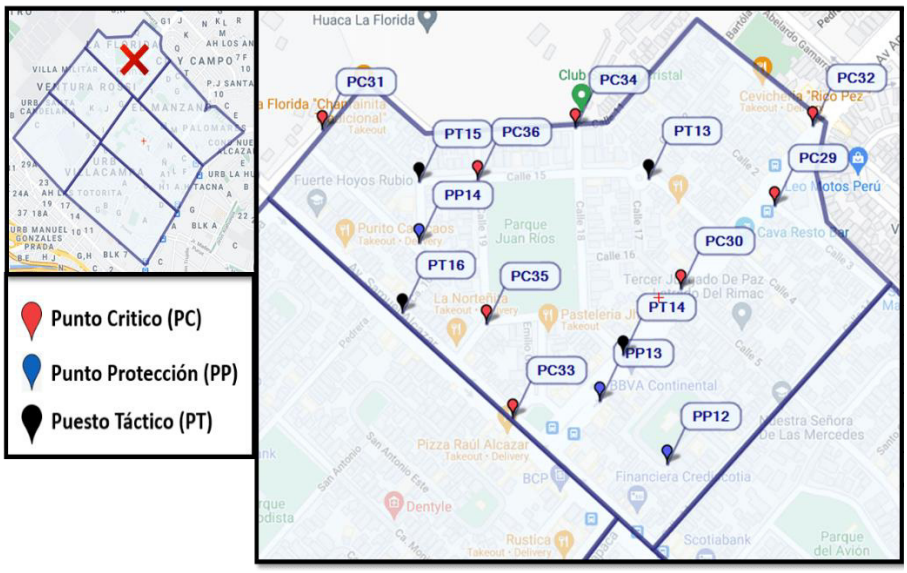
Los dos primeros escenarios plantean visitar de diez a doce puntos en un solo recorrido, este tipo de decisión no es muy frecuente, los algoritmos matemáticos logran reducir en promedio 55% y 44% las distancias de las rutas empíricas construidas para el primer y segundo escenario respectivamente.

Los dos últimos escenarios son decisiones cotidianas que se puedan dar durante el patrullaje, esto se debe a que en estos escenarios se plantean visitar de cinco a siete puntos, el uso de los algoritmos matemáticos logra reducir en promedio 50% y 42% las distancias de recorrido en las rutas empíricas construidas para el tercer y cuarto escenario respectivamente.

Análisis para el sector S5

Figura 26

Mapa del sector S5 de CPNP Manzano



Nota. La leyenda se encuentra en Anexo 9.3.

La Tabla 28 señala que existen once puntos que son visitados durante los recorridos del patrullaje, además se utilizan cuatro puestos tácticos para el inicio del recorrido.

Tabla 28

Información del sector S5 de CPNP Manzano

Puntos del sector					
Origen	Táctico	PT13	PT14	PT15	PT16
	Índice	1	2	3	4
Destinos	Crítico	PC29	PC30	PC31	PC32
	Índice	1	2	3	4
	Crítico	PC33	PC34	PC35	PC36
	Índice	5	6	7	8
	Protección	PP12	PP13	PP14	
	Índice	9	10	11	

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

De la Tabla 28 se han seleccionado aleatoriamente puntos para construir rutas empíricas. En la Tabla 28.1. se muestran las rutas empíricas.

Tabla 28.1

Datos aleatorios para construir rutas empíricas en el sector S5 de CPNP Manzano

Escenario 1 : 100% (total Puntos) : 11 puntos											
O	Índice:	2									
	Punto Inicial:	PT14									
	Aleatorios:	0.954	0.503	0.124	0.814	0.885	0.431	0.899	0.281	0.322	0.545	0.373
D	Índice:	1	6	11	4	3	7	2	10	9	5	8
	Destino:	PC29	PC34	PP14	PC32	PC31	PC35	PC30	PP13	PP12	PC33	PC36
Escenario 2 : 80% (total Puntos) : 9 puntos											
O	Índice:	1									
	Punto Inicial:	PT13									
	Aleatorios:	0.474	0.416	0.569	0.492	0.371	0.838	0.411	0.842	0.519	0.744	0.834
D	Índice:	8	9	5	7	11	2	10	1	6		
	Destino:	PC36	PP12	PC33	PC35	PP14	PC30	PP13	PC29	PC34		
Escenario 3 : 60% (total Puntos) : 7 puntos											
O	Índice:	4									
	Punto Inicial:	PT16									
	Aleatorios:	0.03	0.634	0.421	0.495	0.157	0.946	0.458	0.65	0.561	0.934	0.095
D	Índice:	11	4	8	6	9	1	7				
	Destino:	PP14	PC32	PC36	PC34	PP12	PC29	PC35				
Escenario 4 : 40% (total Puntos) : 4 puntos											
O	Índice:	3									
	Punto Inicial:	PT15									
	Aleatorios:	0.248	0.729	0.376	0.389	0.94	0.49	0.772	0.427	0.078	0.17	0.682
D	Índice:	9	3	8	7							
	Destino:	PP12	PC31	PC36	PC35							

Nota. Se muestra los resultados aleatorios de las fórmulas de Excel. Se denota que O es origen y D es destinos.

En el primer, segundo tercer y cuarto escenario se visitan once, nueve, siete y cuatro puntos respectivamente.

Los datos en cada escenario han sido usados para presentar los resultados de los algoritmos en la Tabla 28.2.

Tabla 28.2

Resultados de la aplicación de algoritmos matemáticos

Comparación de Resultados : Escenario 1														
Ruta empírica	Ruta:	PT14	PC29	PC34	PP14	PC32	PC31	PC35	PC30	PP13	PP12	PC33	PC36	PT14
	Distancia:	8753	metros											
A. Clarke y Wright	Ruta:	PT14	PP14	PC31	PC36	PC35	PC33	PC34	PP13	PP12	PC29	PC32	PC30	PT14
	Distancia:	4237	metros											
A. Vecino más Cercano	Ruta:	PT14	PP13	PC34	PC33	PC35	PC36	PP14	PC31	PP12	PC30	PC29	PC32	PT14
	Distancia:	4322	metros											
A. Búsqueda Tabú	Ruta:	PT14	PP13	PC33	PC35	PP14	PC31	PC36	PC34	PC30	PC32	PC29	PP12	PT14
	Distancia:	3947	metros											
A. Genético	Ruta:	PT14	PC30	PC32	PC29	PP12	PC33	PC35	PP14	PC31	PC36	PC34	PP13	PT14
	Distancia:	4060	metros											
Comparación de Resultados : Escenario 2														
Ruta empírica	Ruta:	PT13	PC36	PP12	PC33	PC35	PP14	PC30	PP13	PC29	PC34	PT13		
	Distancia:	5481	metros											
A. Clarke y Wright	Ruta:	PT13	PP14	PC36	PC35	PC33	PC34	PP13	PP12	PC29	PC30	PT13		
	Distancia:	3189	metros											
A. Vecino más Cercano	Ruta:	PT13	PC30	PP12	PP13	PC34	PC33	PC35	PC36	PP14	PC29	PT13		
	Distancia:	3476	metros											
A. Búsqueda Tabú	Ruta:	PT13	PC34	PC36	PP14	PC35	PC33	PP13	PP12	PC29	PC30	PT13		
	Distancia:	2866	metros											
A. Genético	Ruta:	PT13	PC34	PC36	PP14	PC35	PC33	PP13	PP12	PC29	PC30	PT13		
	Distancia:	2866	metros											
Comparación de Resultados : Escenario 3														
Ruta empírica	Ruta:	PT16	PP14	PC32	PC36	PC34	PP12	PC29	PC35	PT16				
	Distancia:	5612	metros											
A. Clarke y Wright	Ruta:	PT16	PC32	PC29	PP12	PC34	PC35	PC36	PP14	PT16				
	Distancia:	3473	metros											
A. Vecino más Cercano	Ruta:	PT16	PP14	PC36	PC35	PC34	PP12	PC29	PC32	PT16				
	Distancia:	3473	metros											
A. Búsqueda Tabú	Ruta:	PT16	PP14	PC36	PC34	PC32	PC29	PP12	PC35	PT16				
	Distancia:	3258	metros											
A. Genético	Ruta:	PT16	PC35	PP12	PC29	PC32	PC34	PC36	PP14	PT16				
	Distancia:	3258	metros											

Comparación de Resultados : Escenario 4							
Ruta empírica	Ruta:	PT15	PP12	PC31	PC36	PC35	PT15
	Distancia:	3330	metros				
A. Clarke y Wright	Ruta:	PT15	PP12	PC35	PC36	PC31	PT15
	Distancia:	2771	metros				
A. Vecino más Cercano	Ruta:	PT15	PC36	PC35	PC31	PP12	PT15
	Distancia:	3324	metros				
A. Búsqueda Tabú	Ruta:	PT15	PC36	PP12	PC35	PC31	PT15
	Distancia:	2712	metros				
A. Genético	Ruta:	PT15	PC31	PC35	PP12	PC36	PT15
	Distancia:	2712	metros				

La comparación porcentual de los resultados se ve en Tabla 28.3.

Tabla 28.3

Comparación porcentual de los resultados

	Escenario 1		Escenario 2		Escenario 3		Escenario 4	
	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual
Ruta empírica	8753	---	5481	---	5612	---	3330	---
A. Clarke y Wright	4237	51.6%	3189	41.8%	3473	38.1%	2771	17%
A. Vecino más Cercano	4322	50.6%	3476	36.6%	3473	38.1%	3324	0.0001
A. Búsqueda Tabú	3947	54.9%	2866	47.7%	3258	41.9%	2712	19%
A. Genético	4060	53.6%	2866	47.7%	3258	41.9%	2712	19%

El primer escenario plantea la situación en que se decida visitar once puntos en un solo recorrido, esta decisión no es elegida cotidianamente, aun así, los algoritmos logran reducir en promedio un 53% la distancia de recorrido.

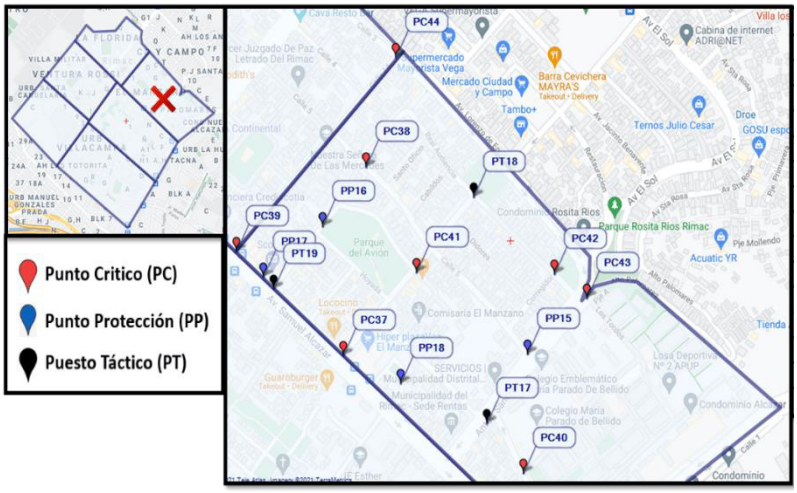
Para los escenarios dos y tres, estas decisiones sí son realizadas porque los recorridos consisten en visitar de siete a nueve puntos y son decisiones generalmente cotidianas, el uso de los algoritmos matemáticos logran reducir en promedio 43% y 40% las distancias de recorrido para cada ruta empírica.

En el cuarto escenario, esta decisión no se elige cotidianamente porque son recorridos cortos, aun así, los algoritmos matemáticos logran reducir en promedio un 14% la distancia de recorrido de los vehículos en la ruta empírica.

Análisis para el sector S6

Figura 27

Mapa del sector S6 de CPNP Manzano



Nota. La leyenda se encuentra en Anexo 9.3.

En la Tabla 29 se muestra que existen doce puntos que son considerados a visitar durante el patrullaje, además se utilizan tres puestos tácticos para el inicio del recorrido.

Tabla 29

Información del sector S6 de CPNP Manzano

		Puntos del sector				
Origen	Táctico	PT17	PT18	PT19		
	Índice	1	2	3		
Destinos	Crítico	PC37	PC38	PC39	PC40	PC41
	Índice	1	2	3	4	5
	Crítico	PC42	PC43	PC44		
	Índice	6	7	8		
	Protección	PP15	PP16	PP17	PP18	
	Índice	9	10	11	12	

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

Con los datos de la Tabla 29 se han seleccionado aleatoriamente los puntos para construir rutas empíricas. Se muestra en la Tabla 29.1.

Tabla 29.1

Datos aleatorios para construir rutas empíricas en el S6 de CPNP Manzano

Escenario 1 : 100% (total Puntos) : 12 puntos													
O	Índice:	3											
	Punto Inicial:	PT19											
D	Aleatorios:	0.16	0.3	0.52	0.306	0.423	0.58	0.252	0.812	0.794	0.761	0.223	0.591
	Índice:	12	9	6	8	7	5	10	1	2	3	11	4
	Destino:	PP18	PP15	PC42	PC44	PC43	PC41	PP16	PC37	PC38	PC39	PP17	PC40
Escenario 2 : 80% (total Puntos) : 10 puntos													
O	Índice:	2											
	Punto Inicial:	PT18											
D	Aleatorios:	0.467	0.429	0.657	0.705	0.085	0.969	0.798	0.167	0.34	0.17	0.03	0.543
	Índice:	6	7	4	3	11	1	2	10	8	9		
	Destino:	PC42	PC43	PC40	PC39	PP17	PC37	PC38	PP16	PC44	PP15		
Escenario 3 : 60% (total Puntos) : 7 puntos													
O	Índice:	2											
	Punto Inicial:	PT18											
D	Aleatorios:	0.814	0.261	0.396	0.915	0.231	0.729	0.655	0.46	0.297	0.882	0.717	0.17
	Índice:	3	10	8	1	11	4	6					
	Destino:	PC39	PP16	PC44	PC37	PP17	PC40	PC42					
Escenario 4 : 40% (total Puntos) : 5 puntos													
O	Índice:	3											
	Punto Inicial:	PT19											
D	Aleatorios:	0.784	0.698	0.99	0.869	0.241	0.55	0.754	0.895	0.96	0.51	0.252	0.394
	Índice:	5	7	1	4	12							
	Destino:	PC41	PC43	PC37	PC40	PP18							

Nota. Se muestra los resultados aleatorios de las fórmulas de Excel. Se denota que O es origen y D es destinos.

Según en la Tabla 29.1, se construyeron cuatro rutas empíricas.

Los puntos de cada escenario han sido usados para presentar los resultados de los algoritmos en la Tabla 29.2.

Tabla 29.2

Resultados de la aplicación de algoritmos matemáticos

Comparación de Resultados : Escenario 1															
Ruta empírica	Ruta:	PT19	PP18	PP15	PC42	PC44	PC43	PC41	PP16	PC37	PC38	PC39	PP17	PC40	PT19
	Distancia:	9035	metros												
A. Clarke y Wright	Ruta:	PT19	PC43	PC42	PP15	PC40	PC41	PP18	PC44	PC38	PC37	PP16	PC39	PP17	PT19
	Distancia:	4603	metros												
A. Vecino más Cercano	Ruta:	PT19	PP17	PC39	PP16	PC37	PC38	PC44	PC41	PP18	PC40	PP15	PC42	PC43	PT19
	Distancia:	4561	metros												
A. Búsqueda Tabú	Ruta:	PT19	PP17	PC39	PP16	PC38	PC44	PC41	PP15	PC42	PC43	PC40	PP18	PC37	PT19
	Distancia:	4106	metros												
A. Genético	Ruta:	PT19	PP17	PC39	PP16	PC38	PC41	PP18	PC40	PC43	PC42	PP15	PC44	PC37	PT19
	Distancia:	4388	metros												
Comparación de Resultados : Escenario 2															
Ruta empírica	Ruta:	PT18	PC42	PC43	PC40	PC39	PP17	PC37	PC38	PP16	PC44	PP15	PT18		
	Distancia:	5240	metros												
A. Clarke y Wright	Ruta:	PT18	PP17	PC39	PP16	PC37	PC38	PC44	PC40	PC43	PC42	PP15	PT18		
	Distancia:	4296	metros												
A. Vecino más Cercano	Ruta:	PT18	PP15	PC40	PC42	PC43	PC44	PC38	PC37	PP16	PP17	PC39	PT18		
	Distancia:	4386	metros												
A. Búsqueda Tabú	Ruta:	PT18	PP15	PC42	PC43	PC40	PC37	PP17	PC39	PP16	PC38	PC44	PT18		
	Distancia:	4025	metros												
A. Genético	Ruta:	PT18	PC44	PC38	PP16	PC39	PP17	PC37	PC40	PC43	PC42	PP15	PT18		
	Distancia:	4025	metros												
Comparación de Resultados : Escenario 3															
Ruta empírica	Ruta:	PT18	PC39	PP16	PC44	PC37	PP17	PC40	PC42	PT18					
	Distancia:	5217	metros												
A. Clarke y Wright	Ruta:	PT18	PP17	PC39	PP16	PC37	PC44	PC40	PC42	PT18					
	Distancia:	3922	metros												
A. Vecino más Cercano	Ruta:	PT18	PC40	PC42	PC44	PC37	PP16	PP17	PC39	PT18					
	Distancia:	3948	metros												
A. Búsqueda Tabú	Ruta:	PT18	PC42	PC40	PC37	PP17	PC39	PP16	PC44	PT18					
	Distancia:	3652	metros												
A. Genético	Ruta:	PT18	PC42	PC40	PC37	PP17	PC39	PP16	PC44	PT18					
	Distancia:	3652	metros												

Comparación de Resultados : Escenario 4								
Ruta empírica	Ruta:	PT19	PC41	PC43	PC37	PC40	PP18	PT19
	Distancia:	5382	metros					
A. Clarke y Wright	Ruta:	PT19	PC43	PC40	PC41	PP18	PC37	PT19
	Distancia:	3564	metros					
A. Vecino más Cercano	Ruta:	PT19	PC37	PP18	PC41	PC40	PC43	PT19
	Distancia:	3564	metros					
A. Búsqueda Tabú	Ruta:	PT19	PC37	PP18	PC40	PC43	PC41	PT19
	Distancia:	3469	metros					
A. Genético	Ruta:	PT19	PC41	PC43	PC40	PP18	PC37	PT19
	Distancia:	3469	metros					

Se ve en la Tabla 29.2 que las rutas diseñados con los algoritmos matemáticos son mejores. Ver la Tabla 29.3.

Tabla 29.3

Comparación porcentual de los resultados

	Escenario 1		Escenario 2		Escenario 3		Escenario 4	
	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual
Ruta empírica	9035	---	5240	---	5217	---	5382	---
A. Clarke y Wright	4603	49.1%	4296	18%	3922	24.8%	3564	34%
A. Vecino más Cercano	4561	49.5%	4386	16.3%	3948	24.3%	3564	34%
A. Búsqueda Tabú	4106	54.6%	4025	23.2%	3652	30%	3469	36%
A. Genético	4388	51.4%	4025	23.2%	3652	30%	3469	36%

Los dos primeros escenarios indica realizar un recorrido para visitar de diez a doce puntos, no es frecuente esta decisión, sin embargo con los algoritmos hubo una reducción en promedio de 51% y 20% en las distancias de recorrido.

Los dos últimos escenarios involucran trayectorias que visiten de cinco a siete puntos, estos son comunes en el patrullaje vehicular, cuando se utiliza los algoritmos matemáticos se logra una reducción en promedio de 27% y 35% en las distancias de recorrido de las rutas empíricas construidas.

En la Tabla 30 se compara las distancias de las rutas cuando se toma una decisión empírica y cuando se toma una decisión en base a técnicas matemáticas.

Tabla 30

Resultados finales para la CPNP Manzano

Sectores	Escenarios	Distancia de la ruta empírica (metros)	Promedio de distancias de rutas con Heurísticas y Metaheurísticas (metros)	Porcentaje de reducción	Estado
S1	No se realiza patrullaje móvil				
S2	Escenario 1	13548	6531.25	51.79%	Reducción
	Escenario 2	7878	5935.75	24.65%	Reducción
	Escenario 3	12242	6345.75	48.16%	Reducción
	Escenario 4	3821	3612	5.47%	Reducción
S3	Escenario 1	8657	4163.25	51.91%	Reducción
	Escenario 2	7255	4336.75	40.22%	Reducción
	Escenario 3	5646	3281.5	41.88%	Reducción
	Escenario 4	2527	2331.5	7.74%	Reducción
S4	Escenario 1	13545	6105.25	54.93%	Reducción
	Escenario 2	9397	5217	44.48%	Reducción
	Escenario 3	10083	5016.5	50.25%	Reducción
	Escenario 4	8238	4804.5	41.68%	Reducción
S5	Escenario 1	8753	4141.5	52.68%	Reducción
	Escenario 2	5481	3099.25	43.45%	Reducción
	Escenario 3	5612	3365.5	40.03%	Reducción
	Escenario 4	3330	2879.75	13.52%	Reducción
S6	Escenario 1	9035	4414.5	51.14%	Reducción
	Escenario 2	5240	4183	20.17%	Reducción
	Escenario 3	5217	3793.5	27.29%	Reducción
	Escenario 4	5382	3516.5	34.66%	Reducción

Nota. La tercera columna puntualiza el efecto de usar técnicas matemáticas sobre decisiones empíricas.

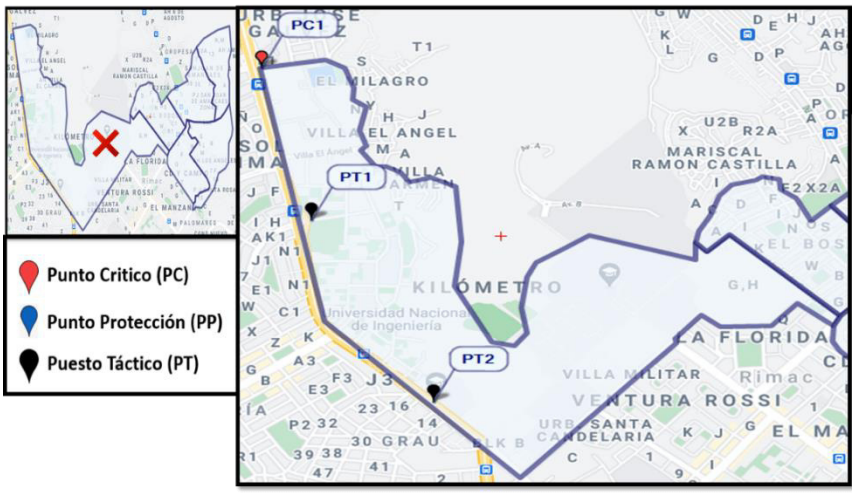
Según la Tabla 30 se puede observar que la aplicación de heurísticas y metaheurísticas para diseñar rutan, han logrado reducir la distancia del recorrido de los vehiculos en las rutinas de patrullaje móvil para la comisaría PNP Manzano.

5.4.4. Análisis de resultados para CPNP Ciudad y Campo

Análisis para el sector S1-1

Figura 28

Mapa del sector S1-1 de CPNP Ciudad y Campo



Nota. La leyenda se encuentra en Anexo 9.4.

Si bien es cierto el mapa muestra que este es el sector más extenso en comparación a los demás, existe solo dos avenidas principales por donde el vehículo puede desplazarse, en esta medida la decisión recurrente es el patrullaje táctico para que se realice presencia policial durante toda la jornada del día. En la Tabla 31 se encuentra la información de los puestos tácticos y puntos críticos.

Tabla 31

Información del sector S1-1 de CPNP Ciudad y Campo

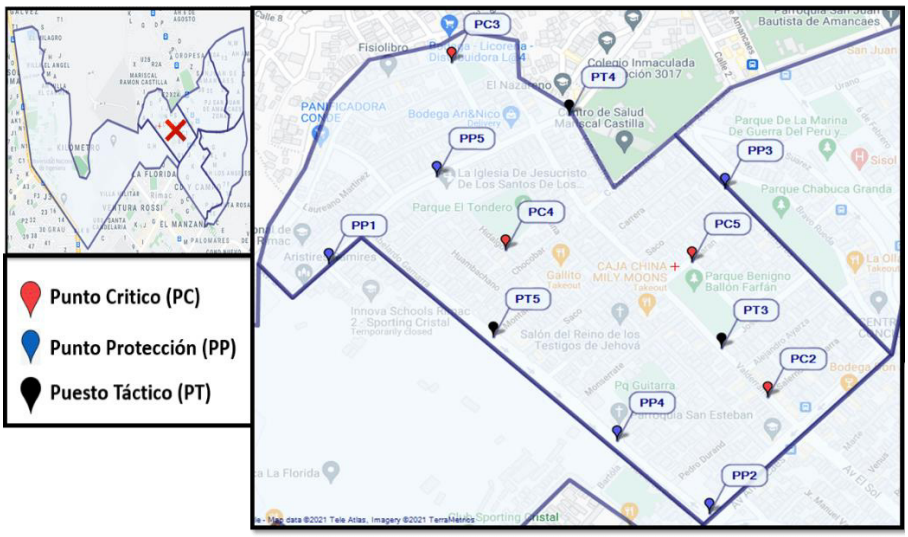
Puntos del sector			
Origen	Táctico	PT1	PT2
	Índice	1	2
Destino	Crítico	PC1	
	Índice	1	

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

Análisis para el sector S1-2

Figura 29

Mapa del sector S1-2 de CPNP Ciudad y Campo



Nota. La leyenda se encuentra en Anexo 9.4.

En la Tabla 32 se presenta la información de los puntos establecidos en este sector. Se indica que hay nueve puntos a custodiar durante los recorridos del vehículo y tres puestos tácticos tomados como punto de partida.

Tabla 32

Información del sector S1-2 de CPNP Ciudad y Campo

Puntos del sector						
Origen	Táctico	PT3	PT4	PT5		
	Índice	1	2	3		
Destinos	Crítico	PC2	PC3	PC4	PC5	
	Índice	1	2	3	4	
	Protección	PP1	PP2	PP3	PP4	PP5
	Índice	5	6	7	8	9

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

Con los datos de la Tabla 32 se han construido rutas empíricas mediante la selección aleatoria de puntos. La selección está en la Tabla 32.1.

Tabla 32.1

Datos aleatorios para construir rutas empíricas en el S1-2 de CPNP Ciudad y Campo

Escenario 1 : 100% (total Puntos) : 9 puntos										
O	Índice:	2								
	Punto Inicial:	PT4								
D	Aleatorios:	0.42	0.893	0.628	0.273	0.143	0.166	0.76	0.294	0.722
	Índice:	5	1	4	7	9	8	2	6	3
	Destino:	PP1	PC2	PC5	PP3	PP5	PP4	PC3	PP2	PC4
Escenario 2 : 80% (total Puntos) : 7 puntos										
O	Índice:	1								
	Punto Inicial:	PT3								
D	Aleatorios:	0.816	0.477	0.803	0.173	0.249	0.678	0.361	0.405	0.137
	Índice:	1	4	2	8	7	3	6		
	Destino:	PC2	PC5	PC3	PP4	PP3	PC4	PP2		
Escenario 3 : 60% (total Puntos) : 5 puntos										
O	Índice:	3								
	Punto Inicial:	PT5								
D	Aleatorios:	0.342	0.825	0.739	0.683	0.118	0.981	0.545	0.126	0.784
	Índice:	7	2	4	5	9				
	Destino:	PP3	PC3	PC5	PP1	PP5				
Escenario 4 : 40% (total Puntos) : 4 puntos										
O	Índice:	3								
	Punto Inicial:	PT5								
D	Aleatorios:	0.194	0.741	0.761	0.475	0.131	0.602	0.498	0.528	0.32
	Índice:	8	2	1	6					
	Destino:	PP4	PC3	PC2	PP2					

Nota. Se muestra los resultados aleatorios de las fórmulas de Excel. Se denota que O es origen y D es destinos.

Según se ve en la Tabla 32.1 se han sido construidos cuatro rutas empíricas. En el primer, segundo, tercer y cuarto escenario se visitan nueve, siete, cinco y cuatro puntos respectivamente.

Se presentan los resultados de los algoritmos matemáticos en la Tabla 32.2, a partir de la información de cada escenario planteado en la Tabla 32.1.

Tabla 32.2

Resultados de la aplicación de algoritmos matemáticos

Comparación de Resultados : Escenario 1												
Ruta empírica	Ruta:	PT4	PP1	PC2	PC5	PP3	PP5	PP4	PC3	PP2	PC4	PT4
	Distancia:	8766	metros									
A. Clarke y Wright	Ruta:	PT4	PP2	PC2	PP3	PC5	PP4	PP1	PP5	PC3	PC4	PT4
	Distancia:	4298	metros									
A. Vecino más Cercano	Ruta:	PT4	PC4	PC3	PP5	PP1	PP4	PC5	PP3	PC2	PP2	PT4
	Distancia:	4298	metros									
A. Búsqueda Tabú	Ruta:	PT4	PC3	PP1	PP5	PC4	PP4	PP2	PC2	PP3	PC5	PT4
	Distancia:	4055	metros									
A. Genético	Ruta:	PT4	PC3	PP5	PP1	PC4	PP4	PP2	PC2	PP3	PC5	PT4
	Distancia:	4076	metros									
Comparación de Resultados : Escenario 2												
Ruta empírica	Ruta:	PT3	PC2	PC5	PC3	PP4	PP3	PC4	PP2	PT3		
	Distancia:	5002	metros									
A. Clarke y Wright	Ruta:	PT3	PC4	PC3	PP4	PC5	PP3	PC2	PP2	PT3		
	Distancia:	3327	metros									
A. Vecino más Cercano	Ruta:	PT3	PC5	PP3	PC2	PP2	PP4	PC4	PC3	PT3		
	Distancia:	3310	metros									
A. Búsqueda Tabú	Ruta:	PT3	PC2	PP2	PP4	PC4	PC3	PC5	PP3	PT3		
	Distancia:	3092	metros									
A. Genético	Ruta:	PT3	PC2	PP3	PC5	PC3	PC4	PP4	PP2	PT3		
	Distancia:	3108	metros									

Comparación de Resultados : Escenario 3								
Ruta empírica	Ruta:	PT5	PP3	PC3	PC5	PP1	PP5	PT5
	Distancia:	5577	metros					
A. Clarke y Wright	Ruta:	PT5	PP3	PC5	PC3	PP5	PP1	PT5
	Distancia:	3545	metros					
A. Vecino más Cercano	Ruta:	PT5	PP5	PC3	PP1	PC5	PP3	PT5
	Distancia:	3731	metros					
A. Búsqueda Tabú	Ruta:	PT5	PP1	PP5	PC3	PP3	PC5	PT5
	Distancia:	3524	metros					
A. Genético	Ruta:	PT5	PP1	PP5	PC3	PP3	PC5	PT5
	Distancia:	3524	metros					
Comparación de Resultados : Escenario 4								
Ruta empírica	Ruta:	PT5	PP4	PC3	PC2	PP2	PT5	
	Distancia:	3987	metros					
A. Clarke y Wright	Ruta:	PT5	PP2	PC2	PP4	PC3	PT5	
	Distancia:	3027	metros					
A. Vecino más Cercano	Ruta:	PT5	PC3	PP4	PP2	PC2	PT5	
	Distancia:	3017	metros					
A. Búsqueda Tabú	Ruta:	PT5	PC3	PC2	PP2	PP4	PT5	
	Distancia:	2969	metros					
A. Genético	Ruta:	PT5	PC3	PC2	PP2	PP4	PT5	
	Distancia:	2969	metros					

En la Tabla 32.2, las rutas diseñadas por los algoritmos matemáticos son mejores que las decididas empíricamente, se ve en las distancias de recorrido. La comparación porcentual se ve en Tabla 32.3.

Tabla 32.3

Comparación porcentual de los resultados

	Escenario 1		Escenario 2		Escenario 3		Escenario 4	
	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual
Ruta empírica	8766	---	5002	---	5577	---	3987	---
A. Clarke y Wright	4298	51%	3327	33.5%	3545	36.4%	3027	24%
A. Vecino más Cercano	4298	51%	3310	33.8%	3731	33.1%	3017	24%
A. Búsqueda Tabú	4055	53.7%	3092	38.2%	3524	36.8%	2969	26%
A. Genético	4076	53.5%	3108	37.9%	3524	36.8%	2969	26%

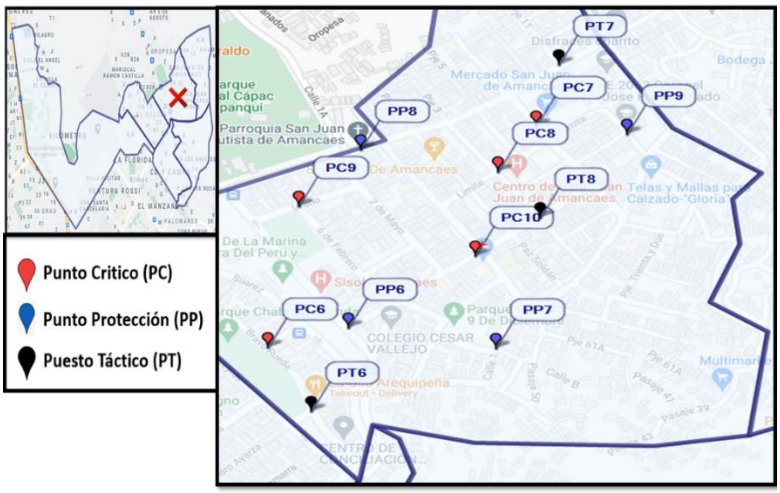
Para los tres primeros escenarios se presenta la situación de realizar un recorrido para visitar de cinco a nueve puntos, con el uso de los algoritmos matemáticos se logró reducir en promedio 52%, 36% y 36% las distancias de recorrido de los vehículos en las rutas empíricas construidas en el primer, segundo y tercer escenario respectivamente.

En el último escenario se presenta la situación de realizar un recorrido que visite cuatro destinos. Esta decisión probablemente no se daría porque son recorridos cortos, pero los algoritmos matemáticos logran reducir en promedio un 25% la distancia de recorrido de los vehículos en la ruta empírica de este escenario.

Análisis para el sector S2-1

Figura 30

Mapa del sector S2-1 de CPNP Ciudad y Campo



Nota. La leyenda se encuentra en Anexo 9.4.

La información mostrada en la Tabla 33 indica que hay nueve puntos que son visitados durante el patrullaje y se utilizan tres puestos tácticos para el inicio del recorrido.

Tabla 33

Información del sector S2-1 de CPNP Ciudad y Campo

Puntos del sector						
Origen	Táctico	PT6	PT7	PT8		
	Índice	1	2	3		
Destinos	Crítico	PC6	PC7	PC8	PC9	PC10
	Índice	1	2	3	4	5
	Protección	PP6	PP7	PP8	PP9	
	Índice	6	7	8	9	

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

Con los datos de la Tabla 33 se han seleccionado aleatoriamente los puntos para construir rutas empíricas. La selección está en la Tabla 33.1.

Tabla 33.1

Datos aleatorios para construir rutas empíricas en el S2-1 de CPNP Ciudad y Campo

Escenario 1 : 100% (total Puntos) : 9 puntos										
O	Índice:	3								
	Punto Inicial:	PT8								
D	Aleatorios:	0.841	0.926	0.982	0.5	0.596	0.396	0.817	0.616	0.12
	Índice:	3	2	1	7	6	8	4	5	9
	Destino:	PC8	PC7	PC6	PP7	PP6	PP8	PC9	PC10	PP9
Escenario 2 : 80% (total Puntos) : 7 puntos										
O	Índice:	1								
	Punto Inicial:	PT6								
D	Aleatorios:	0.696	0.916	0.35	0.837	0.18	0.878	0.797	0.091	0.772
	Índice:	6	1	7	3	8	2	4		
	Destino:	PP6	PC6	PP7	PC8	PP8	PC7	PC9		
Escenario 3 : 60% (total Puntos) : 5 puntos										
O	Índice:	3								
	Punto Inicial:	PT8								
D	Aleatorios:	0.064	0.754	0.948	0.782	0.5	0.094	0.079	0.921	0.903
	Índice:	9	5	1	4	6				
	Destino:	PP9	PC10	PC6	PC9	PP6				
Escenario 4 : 40% (total Puntos) : 4 puntos										
O	Índice:	1								
	Punto Inicial:	PT6								
D	Aleatorios:	0.664	0.932	0.626	0.488	0.849	0.545	0.363	0.338	0.695
	Índice:	4	1	5	7					
	Destino:	PC9	PC6	PC10	PP7					

Nota. Se muestra los resultados aleatorios de las fórmulas de Excel. Se denota que O es origen y D es destinos.

Según la Tabla 33.1, fue posible construir cuatro rutas empíricas. En el primer, segundo, tercer y cuarto escenario se visitan nueve, siete, cinco y cuatro puntos respectivamente.

Los datos acerca del punto de partida y los destinos visitados en cada escenario construidos han sido usados para presentar los resultados de los algoritmos matemáticos en la Tabla 33.2.

Tabla 33.2

Resultados de la aplicación de algoritmos matemáticos

Comparación de Resultados : Escenario 1												
Ruta empírica	Ruta:	PT8	PC8	PC7	PC6	PP7	PP6	PP8	PC9	PC10	PP9	PT8
	Distancia:	4622	metros									
A. Clarke y Wright	Ruta:	PT8	PC9	PC6	PP6	PP8	PC10	PP7	PC8	PC7	PP9	PT8
	Distancia:	3068	metros									
A. Vecino más Cercano	Ruta:	PT8	PC7	PC8	PC10	PP7	PP9	PP8	PP6	PC9	PC6	PT8
	Distancia:	3547	metros									
A. Búsqueda Tabú	Ruta:	PT8	PP7	PP6	PC6	PC9	PP8	PC10	PC8	PC7	PP9	PT8
	Distancia:	2800	metros									
A. Genético	Ruta:	PT8	PP7	PP6	PC6	PC9	PP8	PC10	PC8	PC7	PP9	PT8
	Distancia:	2800	metros									
Comparación de Resultados : Escenario 2												
Ruta empírica	Ruta:	PT6	PP6	PC6	PP7	PC8	PP8	PC7	PC9	PT6		
	Distancia:	3662	metros									
A. Clarke y Wright	Ruta:	PT6	PC8	PC7	PP7	PP8	PC9	PC6	PP6	PT6		
	Distancia:	2521	metros									
A. Vecino más Cercano	Ruta:	PT6	PP6	PP8	PC9	PC6	PP7	PC8	PC7	PT6		
	Distancia:	2771	metros									
A. Búsqueda Tabú	Ruta:	PT6	PC6	PC9	PP8	PC8	PC7	PP7	PP6	PT6		
	Distancia:	2188	metros									
A. Genético	Ruta:	PT6	PP6	PP7	PC7	PC8	PP8	PC9	PC6	PT6		
	Distancia:	2188	metros									

Comparación de Resultados : Escenario 3								
Ruta empírica	Ruta:	PT8	PP9	PC10	PC6	PC9	PP6	PT8
	Distancia:	2754	metros					
A. Clarke y Wright	Ruta:	PT8	PC9	PC6	PP6	PC10	PP9	PT8
	Distancia:	2708	metros					
A. Vecino más Cercano	Ruta:	PT8	PC10	PP6	PC9	PC6	PP9	PT8
	Distancia:	2753	metros					
A. Búsqueda Tabú	Ruta:	PT8	PC10	PP6	PC6	PC9	PP9	PT8
	Distancia:	2699	metros					
A. Genético	Ruta:	PT8	PP9	PC9	PC6	PP6	PC10	PT8
	Distancia:	2699	metros					
Comparación de Resultados : Escenario 4								
Ruta empírica	Ruta:	PT6	PC9	PC6	PC10	PP7	PT6	
	Distancia:	1947	metros					
A. Clarke y Wright	Ruta:	PT6	PP7	PC10	PC9	PC6	PT6	
	Distancia:	1789	metros					
A. Vecino más Cercano	Ruta:	PT6	PC6	PC9	PC10	PP7	PT6	
	Distancia:	1789	metros					
A. Búsqueda Tabú	Ruta:	PT6	PC6	PC9	PC10	PP7	PT6	
	Distancia:	1789	metros					
A. Genético	Ruta:	PT6	PC6	PC9	PC10	PP7	PT6	
	Distancia:	1789	metros					

Los resultados mostrados en la Tabla 33.2 señalan que los algoritmos matemáticos ofrecen mejores rutas en comparación a las decididas empíricamente, se ve reflejado en las distancias de cada ruta. La comparación porcentual se ve en Tabla 33.3.

Tabla 33.3

Comparación porcentual de los resultados

	Escenario 1		Escenario 2		Escenario 3		Escenario 4	
	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual
Ruta empírica	4622	---	3662	---	2754	---	1947	---
A. Clarke y Wright	3068	33.6%	2521	31.2%	2708	1.7%	1789	8%
A. Vecino más Cercano	3547	23.3%	2771	24.3%	2753	0%	1789	8%
A. Búsqueda Tabú	2800	39.4%	2188	40.3%	2699	2%	1789	8%
A. Genético	2800	39.4%	2188	40.3%	2699	2%	1789	8%

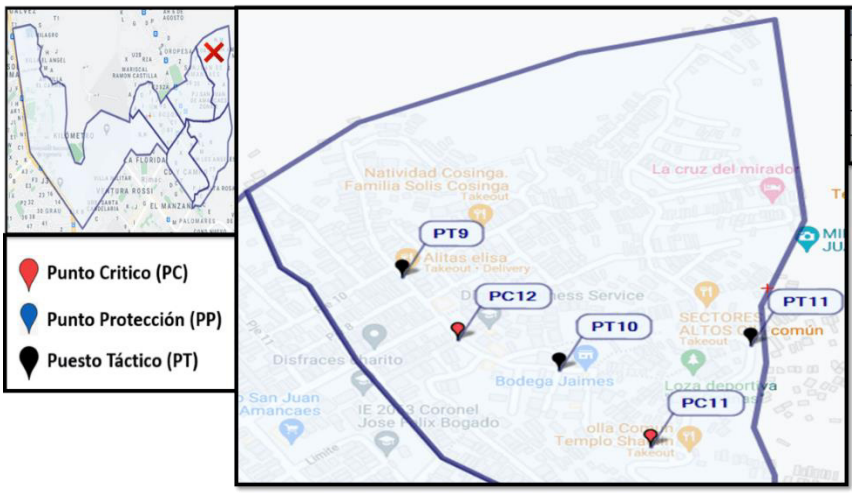
En los tres primeros escenarios, la decisión de visitar de cinco a nueve puntos son comunes, los algoritmos logran reducir en promedio 34%, 34% y 1% en el primer, segundo y tercer escenario respectivamente las distancias de recorrido en cada ruta. Con respecto al tercer escenario, no se aprecia una reducción considerable porque la ruta empírica construida es posiblemente una ruta con recorrido corto.

El último escenario es cuatro destinos en una sola trayectoria, esta decisión no sería la mejor, pues es muy corto el recorrido, los algoritmos logran reducir en promedio 8% la distancia del recorrido de la ruta empírica. No se aprecia una reducción considerable pero se ve una mejora.

Análisis para el sector S2-2

Figura 31

Mapa del sector S6 de CPNP Rímac



Nota. La leyenda se encuentra en Anexo 9.4.

La zona geográfica de este sector dificulta el desplazamiento de los vehículos, gran parte del área esta en los alrededores del cerro Amancaes, por ello un mejor control es que el vehículo permanezca en los puntos realizando el patrullaje táctico. En la Tabla 34 se encuentra la información del sector.

Tabla 34

Información del sector S2-2 de CPNP Ciudad y Campo

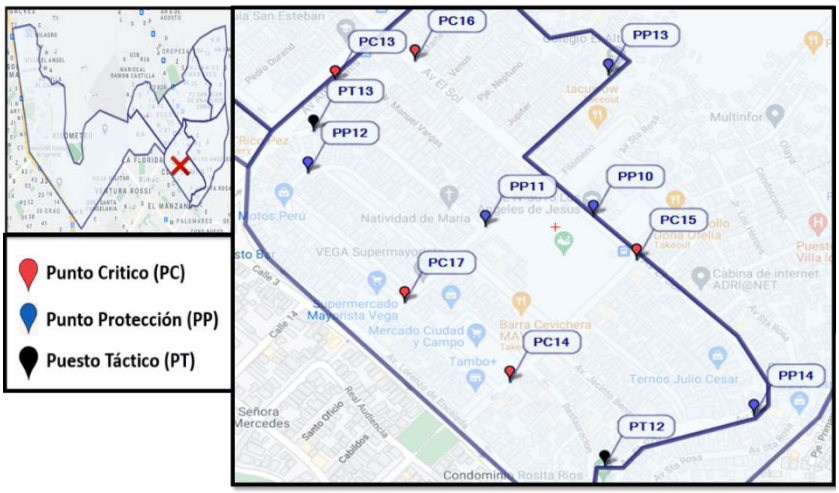
Puntos del sector				
Origen	Táctico	PT9	PT10	PT11
	Índice	1	2	3
Destinos	Crítico	PC11	PC12	
	Índice	1	2	

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

Análisis para el sector S3-1

Figura 32

Mapa del sector S3 de CPNP Manzano



Nota. La leyenda se encuentra en Anexo 9.4.

La información mostrada en la Tabla 35 indica que existen diez puntos que son regularmente visitados durante la rutina de patrullaje móvil, además se utilizan dos puestos tácticos para el inicio del recorrido.

Tabla 35

Información del sector S3-1 de CPNP Ciudad y Campo

		Puntos del sector				
Origen	Táctico	PT12	PT13			
	Índice	1	2			
Destinos	Crítico	PC13	PC14	PC15	PC16	PC17
	Índice	1	2	3	4	5
	Protección	PP10	PP11	PP12	PP13	PP14
	Índice	6	7	8	9	10

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

Con los datos de la Tabla 35 se han construido rutas empíricas mediante la selección aleatoria de puntos. La selección está en la Tabla 35.1.

Tabla 35.1

Datos aleatorios para construir rutas empíricas en el S3-1 de CPNP Ciudad y Campo

Escenario 1 : 100% (total Puntos) : 10 puntos											
O	Índice:	1									
	Punto Inicial:	PT12									
D	Aleatorios:	0.311	0.828	0.085	0.014	0.349	0.964	0.421	0.928	0.27	0.696
	Índice:	7	3	9	10	6	1	5	2	8	4
	Destino:	PP11	PC15	PP13	PP14	PP10	PC13	PC17	PC14	PP12	PC16
Escenario 2 : 80% (total Puntos) : 8 puntos											
O	Índice:	2									
	Punto Inicial:	PT13									
D	Aleatorios:	0.55	0.259	0.489	0.387	0.128	0.068	0.244	0.672	0.639	0.431
	Índice:	3	7	4	6	9	10	8	1		
	Destino:	PC15	PP11	PC16	PP10	PP13	PP14	PP12	PC13		
Escenario 3 : 60% (total Puntos) : 6 puntos											
O	Índice:	2									
	Punto Inicial:	PT13									
D	Aleatorios:	0.205	0.022	0.552	0.183	0.055	0.264	0.41	0.759	0.989	0.062
	Índice:	6	10	3	7	9	5				
	Destino:	PP10	PP14	PC15	PP11	PP13	PC17				
Escenario 4 : 40% (total Puntos) : 4 puntos											
O	Índice:	1									
	Punto Inicial:	PT12									
D	Aleatorios:	0.519	0.199	0.785	0.384	0.199	0.446	0.312	0.808	0.743	0.762
	Índice:	5	10	2	7						
	Destino:	PC17	PP14	PC14	PP11						

Nota. Se muestra los resultados aleatorios de las fórmulas de Excel. Se denota que O es origen y D es destinos.

En la Tabla 35.1, se han construido cuatro rutas empíricas distintas. En el primer, segundo, tercer y cuarto escenario se visitan diez, ocho, seis y cuatro puntos respectivamente.

Se presentan los resultados de los algoritmos matemáticos en la Tabla 35.2.

Tabla 35.2

Resultados de la aplicación de algoritmos matemáticos

Comparación de Resultados : Escenario 1													
Ruta empírica	Ruta:	PT12	PP11	PC15	PP13	PP14	PP10	PC13	PC17	PC14	PP12	PC16	PT12
	Distancia:	6631	metros										
A. Clarke y Wright	Ruta:	PT12	PP12	PC13	PC16	PC17	PP11	PC14	PP13	PP10	PC15	PP14	PT12
	Distancia:	4342	metros										
A. Vecino más Cercano	Ruta:	PT12	PC15	PP10	PP13	PC14	PP11	PC16	PC17	PC13	PP12	PP14	PT12
	Distancia:	4701	metros										
A. Búsqueda Tabú	Ruta:	PT12	PC14	PC17	PP12	PC13	PC16	PP11	PP10	PP13	PC15	PP14	PT12
	Distancia:	3888	metros										
A. Genético	Ruta:	PT12	PC14	PC17	PP12	PC13	PC16	PP11	PP10	PP13	PC15	PP14	PT12
	Distancia:	3888	metros										
Comparación de Resultados : Escenario 2													
Ruta empírica	Ruta:	PT13	PC15	PP11	PC16	PP10	PP13	PP14	PP12	PC13	PT13		
	Distancia:	5808	metros										
A. Clarke y Wright	Ruta:	PT13	PP14	PC15	PP13	PP10	PP11	PC16	PC13	PP12	PT13		
	Distancia:	3895	metros										
A. Vecino más Cercano	Ruta:	PT13	PP12	PC13	PC16	PP11	PP10	PP13	PC15	PP14	PT13		
	Distancia:	3805	metros										
A. Búsqueda Tabú	Ruta:	PT13	PP12	PP11	PP10	PC15	PP14	PP13	PC16	PC13	PT13		
	Distancia:	3715	metros										
A. Genético	Ruta:	PT13	PC13	PC16	PP13	PP14	PC15	PP10	PP11	PP12	PT13		
	Distancia:	3715	metros										

Comparación de Resultados : Escenario 3									
Ruta empírica	Ruta:	PT13	PP10	PP14	PC15	PP11	PP13	PC17	PT13
	Distancia:	4637	metros						
A. Clarke y Wright	Ruta:	PT13	PP14	PC15	PP13	PP10	PP11	PC17	PT13
	Distancia:	3796	metros						
A. Vecino más Cercano	Ruta:	PT13	PC17	PP11	PP10	PP13	PC15	PP14	PT13
	Distancia:	3796	metros						
A. Búsqueda Tabú	Ruta:	PT13	PC17	PP11	PP10	PC15	PP14	PP13	PT13
	Distancia:	3678	metros						
A. Genético	Ruta:	PT13	PC17	PP11	PP10	PC15	PP14	PP13	PT13
	Distancia:	3678	metros						
Comparación de Resultados : Escenario 4									
Ruta empírica	Ruta:	PT12	PC17	PP14	PC14	PP11	PT12		
	Distancia:	3888	metros						
A. Clarke y Wright	Ruta:	PT12	PP11	PC17	PC14	PP14	PT12		
	Distancia:	2856	metros						
A. Vecino más Cercano	Ruta:	PT12	PC14	PP11	PC17	PP14	PT12		
	Distancia:	2891	metros						
A. Búsqueda Tabú	Ruta:	PT12	PC14	PC17	PP11	PP14	PT12		
	Distancia:	2826	metros						
A. Genético	Ruta:	PT12	PC14	PC17	PP11	PP14	PT12		
	Distancia:	2826	metros						

Nota. Se muestra los resultados aleatorios de las fórmulas de Excel.

En la Tabla 35.2 se ve que las soluciones de los algoritmos son mejores, las distancias de recorrido son menores a las trayectorias decididas empíricamente. La comparación porcentual se ve en la Tabla 35.3.

Tabla 35.3

Comparación porcentual de los resultados

	Escenario 1		Escenario 2		Escenario 3		Escenario 4	
	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual
Ruta empírica	6631	---	5808	---	4637	---	3888	---
A. Clarke y Wright	4342	34.5%	3895	32.9%	3796	18.1%	2856	27%
A. Vecino más Cercano	4701	29.1%	3805	34.5%	3796	18.1%	2891	26%
A. Búsqueda Tabú	3888	41.4%	3715	36%	3678	20.7%	2826	27%
A. Genético	3888	41.4%	3715	36%	3678	20.7%	2826	27%

El primer escenario plantea la situación en que se decida visitar diez puntos en un solo recorrido, esta decisión no es tomada en cuenta por ser una trayectoria larga, los algoritmos matemáticos en el diseño de rutas logran reducir en promedio 37% la distancia de recorrido de la ruta empírica.

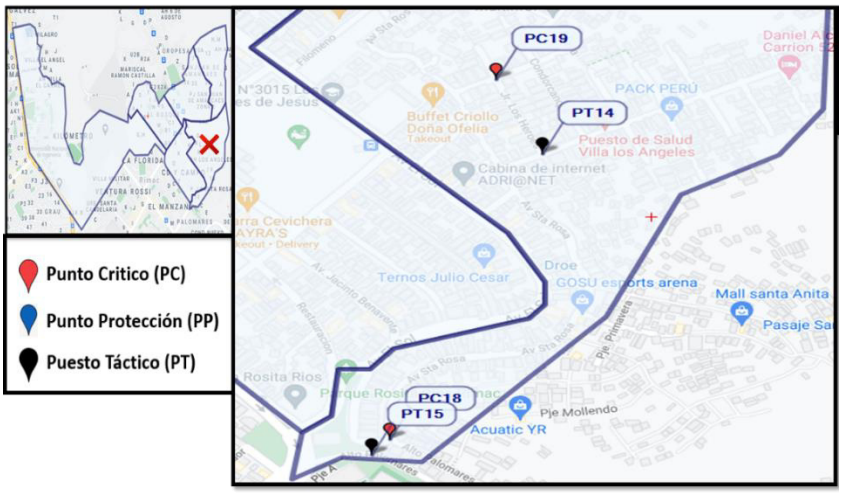
Para los escenarios dos y tres, estas decisiones sí son realizadas porque los recorridos consisten en visitar de seis a ocho puntos y son decisiones generalmente cotidianas, el uso de los algoritmos matemáticos logró reducir en promedio 35% y 19% las distancias de recorrido para cada ruta empírica construida en el segundo y tercer escenario respectivamente.

El último escenario indica un recorrido que visite cuatro puntos, esta decisión no es recurrente porque son rutas cortas, los algoritmos matemáticos logran reducir en promedio un 27% la distancia de recorrido de los vehículos en la ruta empírica de este escenario.

Análisis para el sector S3-2

Figura 33

Mapa del sector S3-2 de CPNP Ciudad y Campo



Nota. La leyenda se encuentra en Anexo 9.4.

En este sector los puntos críticos se encuentran en una avenida principal, en los alrededores se encuentra un conjunto de viviendas donde el tránsito vehicular no es permitido. La medida de protección es de patrullaje táctico, los vehículos se estacionan en los puntos críticos o también los puestos tácticos. La Tabla 36 muestra la información del sector.

Tabla 36

Información del sector S3-2 de CPNP Ciudad y Campo

	Puntos del sector		
Origen	Táctico	PT14	PT15
	Índice	1	2
Destinos	Crítico	PC18	PC19
	Índice	1	2

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

En la Tabla 37 se compara las distancias de las rutas cuando se toma una decisión empírica y cuando se toma una decisión en base a técnicas matemáticas.

Tabla 37

Resultados finales para la CPNP Ciudad y Campo

Sectores	Escenarios	Distancia de la ruta empírica (metros)	Promedio de distancias de rutas con Heurísticas y Metaheurísticas (metros)	Porcentaje de reducción	Estado
S1 - 1	No se realiza patrullaje móvil				
S1 - 2	Escenario 1	8766	4181.5	52.30%	Reducción
	Escenario 2	5002	3209.25	35.84%	Reducción
	Escenario 3	5577	3581	35.79%	Reducción
	Escenario 4	3987	2995.5	24.87%	Reducción
S2 - 1	Escenario 1	4622	3053.75	33.93%	Reducción
	Escenario 2	3662	2417	34.00%	Reducción
	Escenario 3	2754	2714.75	1.43%	Reducción
	Escenario 4	1947	1789	8.12%	Reducción
S2 - 2	No se realiza patrullaje móvil				
S3 - 1	Escenario 1	6631	4204.75	36.59%	Reducción
	Escenario 2	5808	3782.5	34.87%	Reducción
	Escenario 3	4637	3737	19.41%	Reducción
	Escenario 4	3888	2849.75	26.70%	Reducción
S3 - 2	No se realiza patrullaje móvil				

Nota. La tercera columna puntualiza el efecto de usar técnicas matemáticas sobre decisiones empíricas.

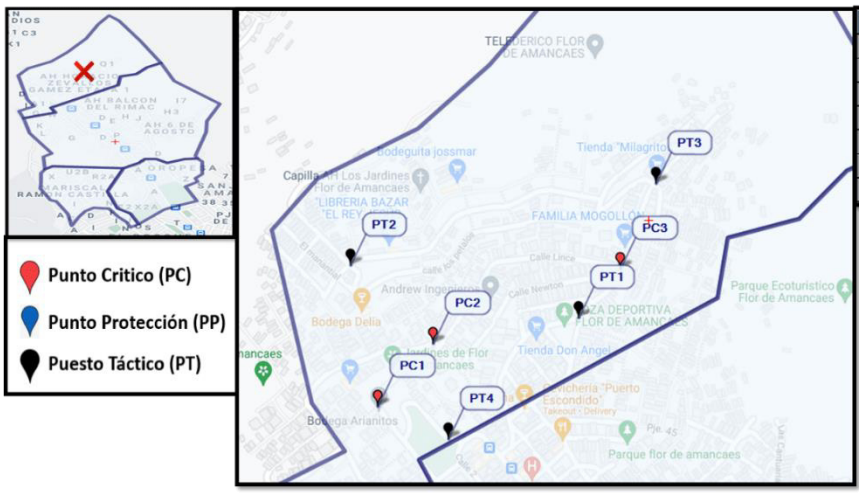
. Según la Tabla 37 se puede observar que la aplicación de heurísticas y metaheurísticas para diseñar rutan, han logrado reducir la distancia del recorrido de los vehiculos en las rutinas de patrullaje móvil para la comisaría PNP Ciudad y Campo

5.4.5. Análisis de resultados para CPNP Flor de Amancaes

Análisis para el sector S1-1

Figura 34

Mapa del sector S1-1 de CPNP Flor de Amancaes



Nota. La leyenda se encuentra en Anexo 9.5.

Este sector presenta pocos caminos por donde se pueda movilizar un vehículo, por lo que la medida de protección es realizar patrullaje táctico. En la Tabla 38 se encuentra la información de los puestos tácticos y puntos críticos que comúnmente el vehículo utiliza en el patrullaje táctico.

Tabla 38

Información del sector S1-1 de CPNP Flor de Amancaes

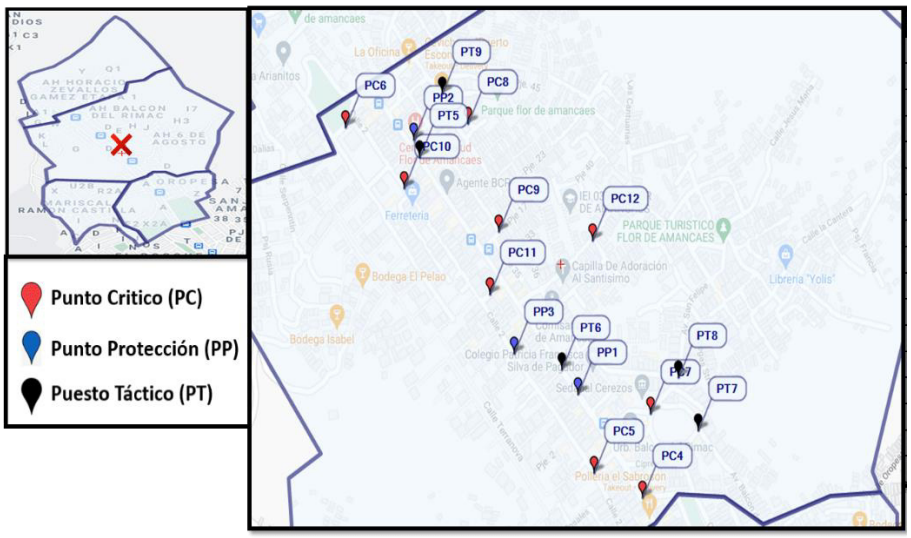
Puntos del sector					
Origen	Táctico	PT1	PT2	PT3	PT4
	Índice	1	2	3	4
Destinos	Crítico	PC1	PC2	PC3	
	Índice	1	2	3	

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

Análisis para el sector S1-2

Figura 35

Mapa del sector S1-2 de CPNP Flor de Amancaes



Nota. La leyenda se encuentra en Anexo 9.5.

La información del sector se ve en la Tabla 39.

Este sector es el más grande de todos, por lo que la medida de protección más efectiva es utilizar dos vehículos para patrullar. Se va a considerar el supuesto de usar solo un vehículo.

Tabla 39

Información del sector S1-2 de CPNP Flor de Amancaes

		Puntos del sector				
Origen	Táctico	PT5	PT6	PT7	PT8	PT9
	Índice	1	2	3	4	5
Destinos	Crítico	PC4	PC5	PC6	PC7	PC8
	Índice	1	2	3	4	5
	Crítico	PC9	PC10	PC11	PC12	
	Índice	6	7	8	9	
	Protección	PP1	PP2	PP3		
	Índice	10	11	12		

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

Se han construido rutas empíricas mediante la selección aleatoria de puntos de la Tabla 39. La selección está en la Tabla 39.1.

Tabla 39.1

Datos aleatorios para construir rutas empíricas en el S1-2 de CPNP Flor de Amancaes

Escenario 1 : 100% (total Puntos) : 12 puntos													
O	Índice:	5											
	Punto Inicial:	PT9											
D	Aleatorios:	0.607	0.906	0.609	0.939	0.229	0.855	0.268	0.391	0.635	0.211	0.532	0.09
	Índice:	6	2	5	1	10	3	9	8	4	11	7	12
	Destino:	PC9	PC5	PC8	PC4	PP1	PC6	PC12	PC11	PC7	PP2	PC10	PP3
Escenario 2 : 80% (total Puntos) : 10 puntos													
O	Índice:	2											
	Punto Inicial:	PT6											
D	Aleatorios:	0.948	0.082	0.659	0.575	0.348	0.287	0.722	0.631	0.196	0.237	0.587	0.939
	Índice:	1	12	4	7	8	9	3	5	11	10		
	Destino:	PC4	PP3	PC7	PC10	PC11	PC12	PC6	PC8	PP2	PP1		
Escenario 3 : 60% (total Puntos) : 7 puntos													
O	Índice:	4											
	Punto Inicial:	PT8											
D	Aleatorios:	0.452	0.104	0.015	0.981	0.43	0.699	0.181	0.629	0.966	0.053	0.991	0.253
	Índice:	6	10	12	2	7	4	9					
	Destino:	PC9	PP1	PP3	PC5	PC10	PC7	PC12					
Escenario 4 : 40% (total Puntos) : 5 puntos													
O	Índice:	1											
	Punto Inicial:	PT5											
D	Aleatorios:	0.676	0.5	0.101	0.428	0.509	0.555	0.087	0.856	0.379	0.81	0.743	0.28
	Índice:	4	7	11	8	6							
	Destino:	PC7	PC10	PP2	PC11	PC9							

Nota. Se muestra los resultados aleatorios de las fórmulas de Excel. Se denota que O es origen y D es destinos.

De la Tabla 39.1 se ve que han sido construidas cuatro rutas empíricas. En el primer, segundo, tercer y cuarto escenario se visitan doce, diez, siete y cinco puntos respectivamente.

Se presentan los resultados de los algoritmos matemáticos en la Tabla 39.2, a partir de la información de cada escenario planteado en la Tabla 39.1.

Tabla 39.2

Resultados de la aplicación de algoritmos matemáticos

Comparación de Resultados : Escenario 1															
Ruta empírica	Ruta:	PT9	PC9	PC5	PC8	PC4	PP1	PC6	PC12	PC11	PC7	PP2	PC10	PP3	PT9
	Distancia:	9038	metros												
A. Clarke y Wright	Ruta:	PT9	PC7	PC4	PC5	PP1	PC12	PP3	PC9	PC11	PC8	PC10	PC6	PP2	PT9
	Distancia:	3678	metros												
A. Vecino más Cercano	Ruta:	PT9	PC8	PC9	PC11	PP3	PP1	PC5	PC12	PC7	PC4	PP2	PC10	PC6	PT9
	Distancia:	3698	metros												
A. Búsqueda Tabú	Ruta:	PT9	PC8	PC9	PC12	PC7	PC4	PC5	PP1	PP3	PC11	PC10	PC6	PP2	PT9
	Distancia:	3306	metros												
A. Genético	Ruta:	PT9	PP2	PC6	PC10	PC11	PP3	PC5	PP1	PC4	PC7	PC12	PC9	PC8	PT9
	Distancia:	3496	metros												
Comparación de Resultados : Escenario 2															
Ruta empírica	Ruta:	PT6	PC4	PP3	PC7	PC10	PC11	PC12	PC6	PC8	PP2	PP1	PT6		
	Distancia:	7184	metros												
A. Clarke y Wright	Ruta:	PT6	PC10	PC6	PP2	PC8	PC11	PP3	PC12	PC7	PC4	PP1	PT6		
	Distancia:	3358	metros												
A. Vecino más Cercano	Ruta:	PT6	PP1	PC12	PC7	PC4	PP3	PC11	PC8	PP2	PC10	PC6	PT6		
	Distancia:	3453	metros												
A. Búsqueda Tabú	Ruta:	PT6	PP1	PC4	PC7	PC12	PC8	PP2	PC6	PC10	PC11	PP3	PT6		
	Distancia:	3264	metros												
A. Genético	Ruta:	PT6	PP1	PC4	PC7	PC12	PC8	PP2	PC6	PC10	PC11	PP3	PT6		
	Distancia:	3264	metros												

Comparación de Resultados : Escenario 3										
Ruta empírica	Ruta:	PT8	PC9	PP1	PP3	PC5	PC10	PC7	PC12	PT8
	Distancia:	5182	metros							
A. Clarke y Wright	Ruta:	PT8	PC10	PC9	PP3	PP1	PC5	PC12	PC7	PT8
	Distancia:	3171	metros							
A. Vecino más Cercano	Ruta:	PT8	PC7	PC5	PP1	PC12	PP3	PC9	PC10	PT8
	Distancia:	3163	metros							
A. Búsqueda Tabú	Ruta:	PT8	PC7	PC5	PP1	PP3	PC10	PC9	PC12	PT8
	Distancia:	2882	metros							
A. Genético	Ruta:	PT8	PC12	PC9	PC10	PP3	PP1	PC5	PC7	PT8
	Distancia:	2882	metros							
Comparación de Resultados : Escenario 4										
Ruta empírica	Ruta:	PT5	PC7	PC10	PP2	PC11	PC9	PT5		
	Distancia:	3448	metros							
A. Clarke y Wright	Ruta:	PT5	PC9	PC7	PC11	PC10	PP2	PT5		
	Distancia:	2583	metros							
A. Vecino más Cercano	Ruta:	PT5	PP2	PC10	PC11	PC9	PC7	PT5		
	Distancia:	2621	metros							
A. Búsqueda Tabú	Ruta:	PT5	PP2	PC10	PC11	PC7	PC9	PT5		
	Distancia:	2583	metros							
A. Genético	Ruta:	PT5	PP2	PC10	PC11	PC7	PC9	PT5		
	Distancia:	2583	metros							

Según se ve en la Tabla 39.2, las rutas diseñadas por los algoritmos matemáticos son mejores que las decididas empíricamente, se ve en las distancias de recorrido. La comparación porcentual se ve en Tabla 39.3.

Tabla 39.3

Comparación porcentual de los resultados

	Escenario 1		Escenario 2		Escenario 3		Escenario 4	
	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual
Ruta empírica	9038	---	7184	---	5182	---	3448	---
A. Clarke y Wright	3678	59.3%	3358	53.3%	3171	38.8%	2583	25%
A. Vecino más Cercano	3698	59.1%	3453	51.9%	3163	39%	2621	24%
A. Búsqueda Tabú	3306	63.4%	3264	54.6%	2882	44.4%	2583	25%
A. Genético	3496	61.3%	3264	54.6%	2882	44.4%	2583	25%

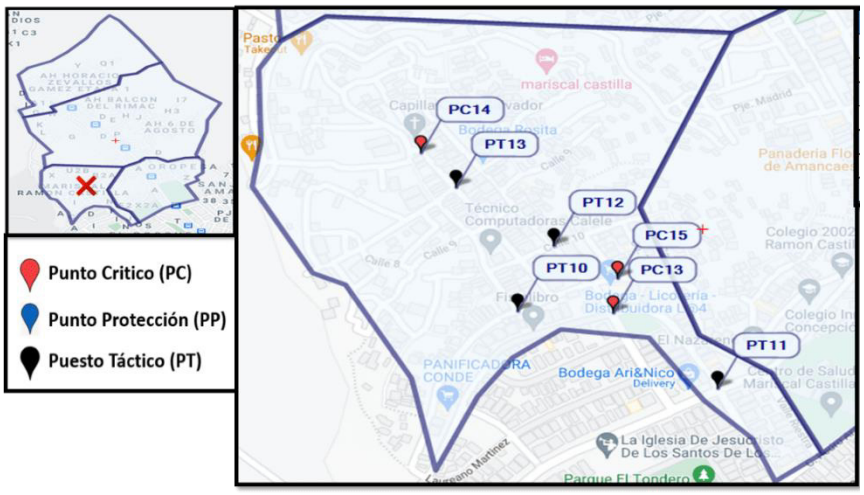
Los dos primeros escenarios se presentan la decisión de realizar un recorrido para visitar de diez a doce puntos, si bien es cierto esta decisión no es elegida cotidianamente, cuando se usó los algoritmos matemáticos, se logró reducir en promedio 61% y 54% las distancias de las rutas empíricas construidas en el primer y segundo escenario.

Para los escenarios tres y cuatro, estas decisiones sí son realizadas porque los recorridos consisten en visitar de cinco a siete puntos y son decisiones generalmente cotidianas, el uso de los algoritmos logró reducir en promedio 42% y 25% en el tercer y cuarto las distancias de recorrido de los vehículos para cada ruta empírica construida.

Análisis para el sector S2-1

Figura 36

Mapa del sector S2-1 de CPNP Flor de Amancaes



Nota. La leyenda se encuentra en Anexo 9.5.

En este sector se realiza el patrullaje de tipo táctico. La decisión de realizar patrullaje táctico se debe a que geográficamente solo hay tres avenidas por donde puede desplazarse el vehículo y en ellas están ubicados la mayoría de los puntos a vigilar, por ello un mejor control es que el vehículo permanezca estacionado cerca a esos puntos. La información es mostrada en la Tabla 40.

Tabla 40

Información del sector S2-1 de CPNP Flor de Amancaes

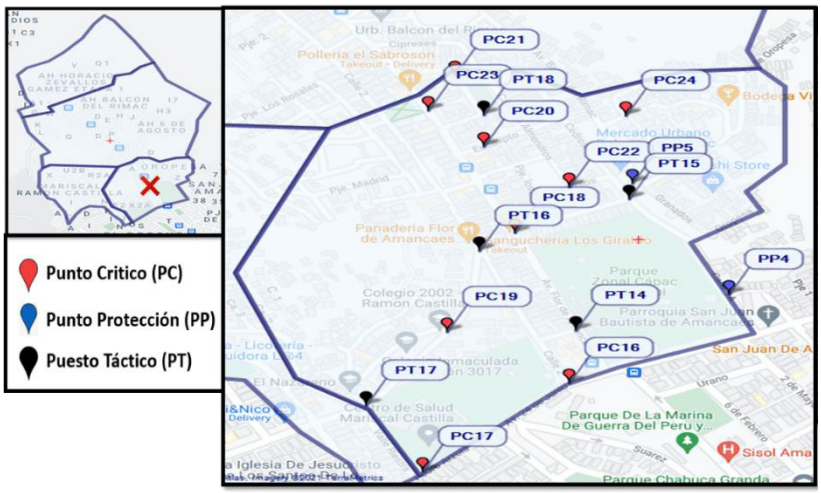
Puntos del sector					
Origen	Táctico	PT10	PT11	PT12	PT13
	Índice	1	2	3	4
Destinos	Crítico	PC13	PC14	PC15	
	Índice	1	2	3	

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

Análisis para el sector S2-2

Figura 37

Mapa del sector S2-2 de CPNP Flor de Amancaes



Nota. La leyenda se encuentra en Anexo 9.5.

En la Tabla 41 se expone que existen once puntos que son considerados a visitar durante el patrullaje, además se utilizan cinco puestos tácticos para el inicio del recorrido.

Tabla 41

Información del sector S2-2 de CPNP Flor de Amancaes

		Puntos del sector				
Origen	Táctico	PT14	PT15	PT16	PT17	PT18
	Índice	1	2	3	4	5
Destinos	Crítico	PC16	PC17	PC18	PC19	
	Índice	1	2	3	4	
	Crítico	PC20	PC21	PC22	PC23	
	Índice	5	6	7	8	
	Protección	PC24	PP4	PP5		
	Índice	9	10	11		

Nota. La información es del periodo de tiempo indicado en Población y Muestra.

De los datos de la Tabla 41 se han seleccionado aleatoriamente los puntos para construir rutas empíricas. Se muestra en la Tabla 41.1.

Tabla 41.1

Datos aleatorios para construir rutas empíricas en el S2-2 de CPNP Flor de Amancaes

Escenario 1 : 100% (total Puntos) : 11 puntos												
O	Índice:	2										
	Punto Inicial:	PT15										
D	Aleatorios:	0.393	0.859	0.3	0.601	0.771	0.362	0.712	0.423	0.694	0.778	0.787
	Índice:	9	1	11	7	4	10	5	8	6	3	2
	Destino:	PC24	PC16	PP5	PC22	PC19	PP4	PC20	PC23	PC21	PC18	PC17
Escenario 2 : 80% (total Puntos) : 9 puntos												
O	Índice:	4										
	Punto Inicial:	PT17										
D	Aleatorios:	0.628	0.983	0.228	0.306	0.463	0.268	0.185	0.781	0.138	0.875	0.966
	Índice:	5	1	9	7	6	8	10	4	11		
	Destino:	PC20	PC16	PC24	PC22	PC21	PC23	PP4	PC19	PP5		
Escenario 3 : 60% (total Puntos) : 7 puntos												
O	Índice:	5										
	Punto Inicial:	PT18										
D	Aleatorios:	0.035	0.083	0.972	0.56	0.415	0.469	0.378	0.244	0.692	0.71	0.392
	Índice:	11	10	1	4	6	5	8				
	Destino:	PP5	PP4	PC16	PC19	PC21	PC20	PC23				
Escenario 4 : 40% (total Puntos) : 4 puntos												
O	Índice:	3										
	Punto Inicial:	PT16										
D	Aleatorios:	0.635	0.802	0.331	0.434	0.601	0.173	0.301	0.134	0.976	0.89	0.481
	Índice:	4	3	8	7							
	Destino:	PC19	PC18	PC23	PC22							

Nota. Se muestra los resultados aleatorios de las fórmulas de Excel. Se denota que O es origen y D es destinos.

En la Tabla 41.1 se aprecia que se construyeron cuatro rutas empíricas. En el primer, segundo, tercer y cuarto escenario se visitan once, nueve, siete y cuatro puntos respectivamente.

Los puntos seleccionados en cada escenario han sido usados para presentar los resultados de los algoritmos matemáticos en la Tabla 41.2.

Tabla 41.2

Resultados de la aplicación de algoritmos matemáticos

Comparación de Resultados : Escenario 1														
Ruta empírica	Ruta:	PT15	PC24	PC16	PP5	PC22	PC19	PP4	PC20	PC23	PC21	PC18	PC17	PT15
	Distancia:	4933	metros											
A. Clarke y Wright	Ruta:	PT15	PC19	PC17	PC23	PC21	PC20	PC18	PC16	PC22	PC24	PP4	PP5	PT15
	Distancia:	3004	metros											
A. Vecino más Cercano	Ruta:	PT15	PP5	PC24	PC22	PC18	PC20	PC21	PC23	PC19	PC17	PC16	PP4	PT15
	Distancia:	2694	metros											
A. Búsqueda Tabú	Ruta:	PT15	PP5	PC24	PC22	PC18	PC20	PC21	PC23	PC19	PC17	PC16	PP4	PT15
	Distancia:	2694	metros											
A. Genético	Ruta:	PT15	PC18	PC20	PC21	PC23	PC19	PC17	PC16	PC22	PP5	PC24	PP4	PT15
	Distancia:	2938	metros											
Comparación de Resultados : Escenario 2														
Ruta empírica	Ruta:	PT17	PC20	PC16	PC24	PC22	PC21	PC23	PP4	PC19	PP5	PT17		
	Distancia:	5509	metros											
A. Clarke y Wright	Ruta:	PT17	PP5	PP4	PC24	PC22	PC16	PC20	PC21	PC23	PC19	PT17		
	Distancia:	3244	metros											
A. Vecino más Cercano	Ruta:	PT17	PC19	PC20	PC21	PC23	PC22	PC24	PP5	PC16	PP4	PT17		
	Distancia:	3602	metros											
A. Búsqueda Tabú	Ruta:	PT17	PC19	PC16	PP4	PP5	PC24	PC22	PC20	PC21	PC23	PT17		
	Distancia:	2783	metros											
A. Genético	Ruta:	PT17	PC23	PC21	PC20	PC22	PC24	PP5	PP4	PC16	PC19	PT17		
	Distancia:	2783	metros											

Comparación de Resultados : Escenario 3										
Ruta empírica	Ruta:	PT18	PP5	PP4	PC16	PC19	PC21	PC20	PC23	PT18
	Distancia:	2614	metros							
A. Clarke y Wright	Ruta:	PT18	PP5	PP4	PC16	PC19	PC23	PC21	PC20	PT18
	Distancia:	2332	metros							
A. Vecino más Cercano	Ruta:	PT18	PC20	PC21	PC23	PC19	PC16	PP5	PP4	PT18
	Distancia:	2432	metros							
A. Búsqueda Tabú	Ruta:	PT18	PC20	PP5	PP4	PC16	PC19	PC23	PC21	PT18
	Distancia:	2312	metros							
A. Genético	Ruta:	PT18	PC20	PP5	PP4	PC16	PC19	PC23	PC21	PT18
	Distancia:	2312	metros							
Comparación de Resultados : Escenario 4										
Ruta empírica	Ruta:	PT16	PC19	PC18	PC23	PC22	PT16			
	Distancia:	1517	metros							
A. Clarke y Wright	Ruta:	PT16	PC22	PC18	PC23	PC19	PT16			
	Distancia:	1217	metros							
A. Vecino más Cercano	Ruta:	PT16	PC18	PC22	PC19	PC23	PT16			
	Distancia:	1238	metros							
A. Búsqueda Tabú	Ruta:	PT16	PC18	PC22	PC23	PC19	PT16			
	Distancia:	1188	metros							
A. Genético	Ruta:	PT16	PC18	PC22	PC23	PC19	PT16			
	Distancia:	1188	metros							

Como se ve en la Tabla 41.2, los diseños de rutas con los cuatro algoritmos matemáticos son mejores en comparación a las decisiones empíricas porque las distancias de recorridos son menores. La comparación porcentual se ve en la Tabla 41.3.

Tabla 41.3

Comparación porcentual de los resultados

	Escenario 1		Escenario 2		Escenario 3		Escenario 4	
	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual	Distancia (metros)	Reducción porcentual
Ruta empírica	4933	---	5509	---	2614	---	1517	---
A. Clarke y Wright	3004	39.1%	3244	41.1%	2332	10.8%	1217	20%
A. Vecino más Cercano	2694	45.4%	3602	34.6%	2432	7%	1238	18%
A. Búsqueda Tabú	2694	45.4%	2783	49.5%	2312	11.6%	1188	22%
A. Genético	2938	40.4%	2783	49.5%	2312	11.6%	1188	22%

El primer escenario plantea la situación en que se decida visitar once puntos en un solo recorrido, si bien es cierto esta decisión no es muy frecuente, con el aporte de algoritmos matemáticos en la planificación de rutas, se pudo reducir en promedio un 43% la distancia de la ruta empírica construida.

Los dos siguientes escenarios, son decisiones generalmente cotidianas porque los recorridos consisten en visitar de siete a nueve puntos, con el uso de los algoritmos matemáticos en la planificación de rutas, estos logran reducir en promedio 44% y 10% las distancias de recorrido de los vehículos para cada ruta empírica construida.

Con respecto al cuarto escenario, esta decisión probablemente no se daría porque son recorridos cortos, aun así, los algoritmos matemáticos logran reducir en promedio un 21% la distancia de recorrido de los vehículos en la ruta empírica de este escenario.

En la Tabla 42 se compara las distancias de las rutas cuando se toma una decisión empírica y cuando se toma una decisión en base a técnicas matemáticas.

Tabla 42

Resultados finales para la CPNP Flor de Amancaes

Sectores	Escenarios	Distancia de la ruta empírica (metros)	Promedio de distancias de rutas con Heurísticas y Metaheurísticas (metros)	Porcentaje de Reducción	Estado
S1 - 1	No se realiza patrullaje móvil				
S1 - 2	Escenario 1	9038	3544.5	60.78%	Reducción
	Escenario 2	7184	3334.75	53.58%	Reducción
	Escenario 3	5182	3024.5	41.63%	Reducción
	Escenario 4	3448	2592.5	24.81%	Reducción
S2 - 1	No se realiza patrullaje móvil				
S2 - 2	Escenario 1	4933	2832.5	42.58%	Reducción
	Escenario 2	5509	3103	43.67%	Reducción
	Escenario 3	2614	2347	10.21%	Reducción
	Escenario 4	1517	1207.75	20.39%	Reducción

Nota. La tercera columna puntualiza el efecto de usar técnicas matemáticas sobre decisiones empíricas.

Según la Tabla 42 se puede observar que la aplicación de heurísticas y metaheurísticas para diseñar rutan, han logrado reducir la distancia del recorrido de los vehiculos en las rutinas de patrullaje móvil para la comisaría PNP Flor de Amancaes.

5.5. Validación de hipótesis

La hipótesis nula:

$$H_0: \mu_1 = \mu_2$$

La hipótesis especifica o lo que se busca probar:

$$H_1: \mu_2 < \mu_1$$

La muestra que se usa son los siguientes:

$\mu_1 =$ La distancia de cada ruta empírica

$\mu_2 =$ El promedio de las distancias de las rutas diseñadas con las metaheurísticas

Se realizó la prueba de diferencia de media. Lo que se busca es rechazar la hipótesis nula (H_0), para validar la hipótesis alternativa (H_1) y asi determinar que el promedio de la distancia de las rutas hechas con metaheurísticas (los cuatro algoritmos matemáticos usados serán generalizados para llamarse metaheurísticas) es menor a la distancia de las rutas decididas empíricamente.

El programa Minitab requiere de al menos quince datos en cada muestra para realizar la prueba de hipótesis de diferencia de medias, por lo que se va a usar la informacion de la sección de resultados de cada comisaria y si fuese necesario se van a completar hasta alcanzar quince datos.

5.5.1. Validacion de hipótesis general

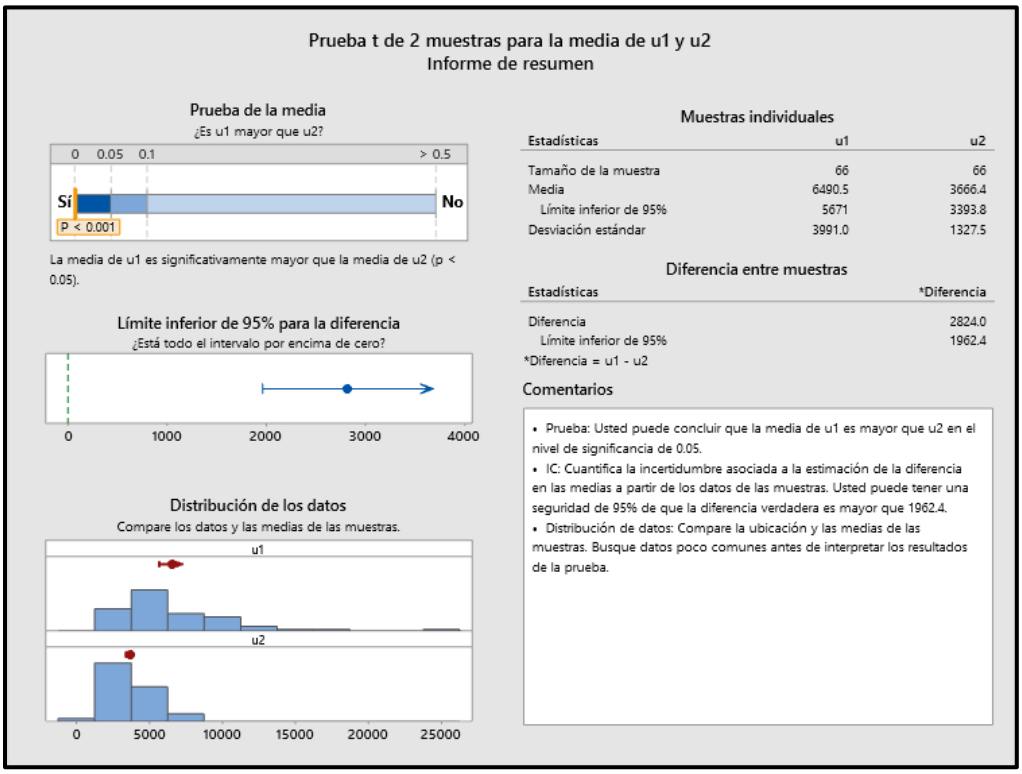
H0: Las rutas diseñadas por metaheurísticas no permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en las jurisdicciones de las comisarías del distrito del Rímac.

H1: Las rutas diseñadas por metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en las jurisdicciones de las comisarías del distrito del Rímac.

Las muestras que se van a utilizar son de los datos de Tabla 16 , Tabla 23, Tabla 30, Tabla 37 y Tabla 42, los resultados del Minitab se ven en la Figura 38.

Figura 38

Resultados del Minitab para la hipótesis general



Según la Figura 38 en la sección de comentarios, se rechaza la hipótesis nula por lo tanto se valida $\mu_2 < \mu_1$ o H1. Se concluye que las rutas diseñadas por metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicciones de las comisarías del distrito del Rimac.

Según lo expuesto en las bases teóricas, las metaheurísticas no encuentran un valor óptimo, pero la solución que brinda es el mejor dentro de todos los intentos de búsqueda que se realizó y esto implicaría una supuesta optimización.

5.5.2. Validación de hipótesis especifica 1

H0: Las rutas diseñadas por metaheurísticas no permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la comisaría PNP Piedra Liza.

H1: Las rutas diseñadas por metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la comisaría PNP Piedra Liza.

Las muestras se ven en la Tabla 43 y los resultados del Minitab en la Figura 39.

Tabla 43

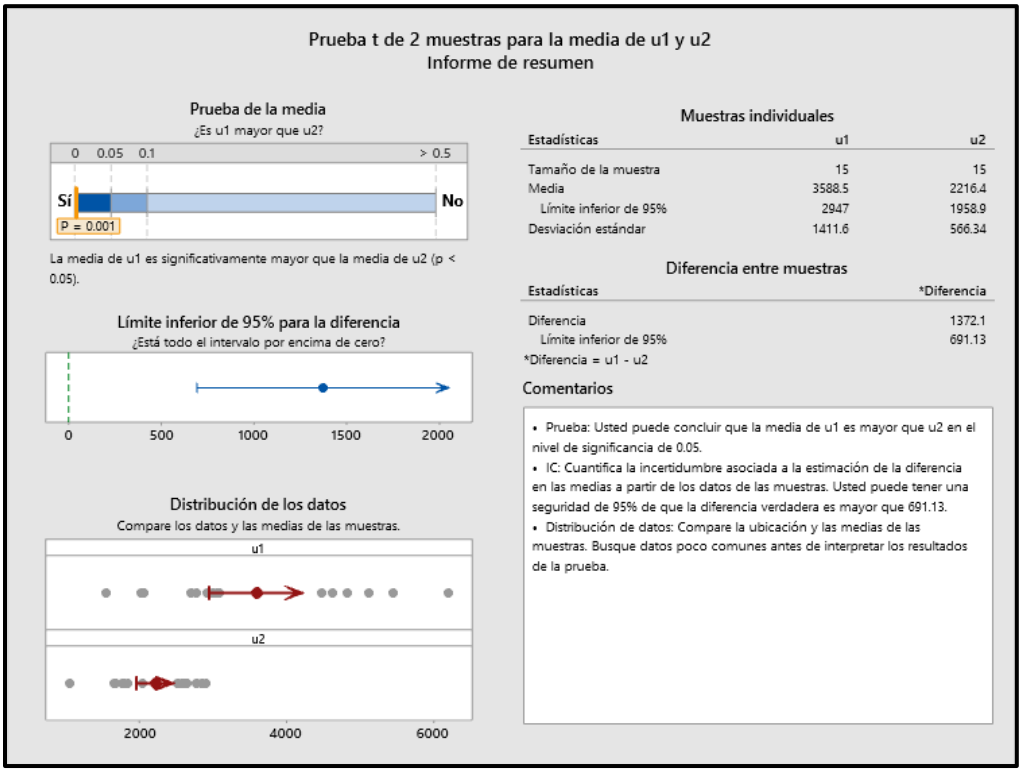
Muestras para la validación de hipotesis en la CPNP Piedra Liza

Datos	Distancia en metros de las rutas							
μ1	5449	5126	4481	1544	2701	2056	2021	4621
	2998	2908	3044	6209	3077	2769	4824	
μ2	2772.5	2839	2649	1039.3	2028.3	1764	1670.8	2648.5
	2511.8	1644.5	1834.5	2898.3	2563	1773.3	2609.8	

Nota. Se usaron datos de la Tabla 16 y añadiendo nuevas soluciones con los procedimientos de la Seccion V.4.

Figura 39

Resultados del Minitab para la hipótesis especifica 1



Según la Figura 39 en la sección de comentarios, se rechaza la hipótesis nula por lo tanto se valida $\mu_2 < \mu_1$ o H1. Se concluye que las rutas diseñadas por metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la CPNP Piedra Liza.

Según lo expuesto en las bases teóricas, las metaheurísticas no encuentran un valor óptimo, pero la solución que brinda es el mejor dentro de todos los intentos de búsqueda que se realizó y esto implicaría una supuesta optimización.

5.5.3. Validación de hipótesis especifica 2

H0: Las rutas diseñadas por metaheurísticas no permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la comisaría PNP Rímac.

H1: Las rutas diseñadas por metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la comisaría PNP Rímac.

Las muestras se ven en la Tabla 44 y los resultados del Minitab en la Figura 40.

Tabla 44

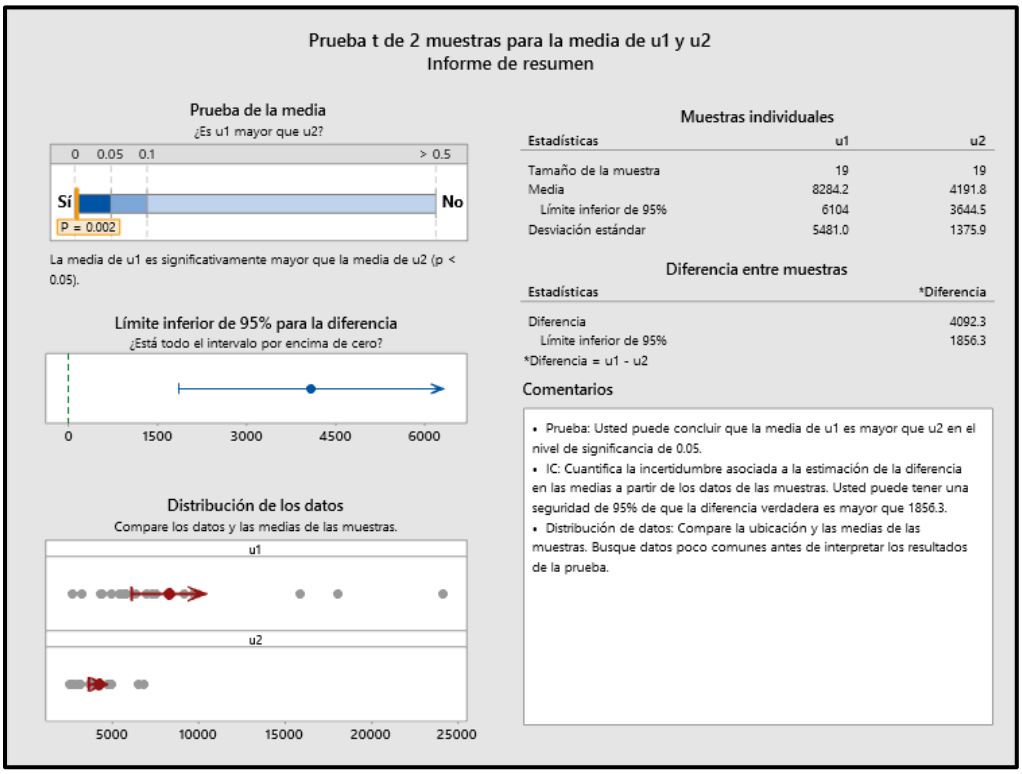
Muestras para la validación de hipotesis en la CPNP Rimac

Datos	Distancia en metros de las rutas									
μ1	7274	5646	4293	4367	6343	7503	9152	4925	24189	15862
	18083	9134	9720	6919	5392	2629	6971	5782	3215	
μ2	3081	3021.5	2640.8	2714.5	4759.5	4682	4950.3	4768	6769.3	6831.3
	6478.5	4691.3	4216.5	4131.3	2990.8	2462.5	3777	3701.3	2977.5	

Nota. Los datos son de los resultados de Tabla 23.

Figura 40

Resultados del Minitab para la hipótesis especifica 2



Según la Figura 40 en la sección de comentarios, se rechaza la hipótesis nula por lo tanto se valida $\mu_2 < \mu_1$ o H1. Se concluye que las rutas diseñadas por metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la CPNP Rímac.

Según lo expuesto en las bases teóricas, las metaheurísticas no encuentran un valor óptimo, pero la solución que brinda es el mejor dentro de todos los intentos de búsqueda que se realizó y esto implicaría una supuesta optimización.

5.5.4. Validación de hipótesis especifica 3

H0: Las rutas diseñadas por metaheurísticas no permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la comisaría PNP Manzano.

H1: Las rutas diseñadas por metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la comisaría PNP Manzano.

Las muestras se ven en la Tabla 45 y los resultados del Minitab en la Figura 41.

Tabla 45

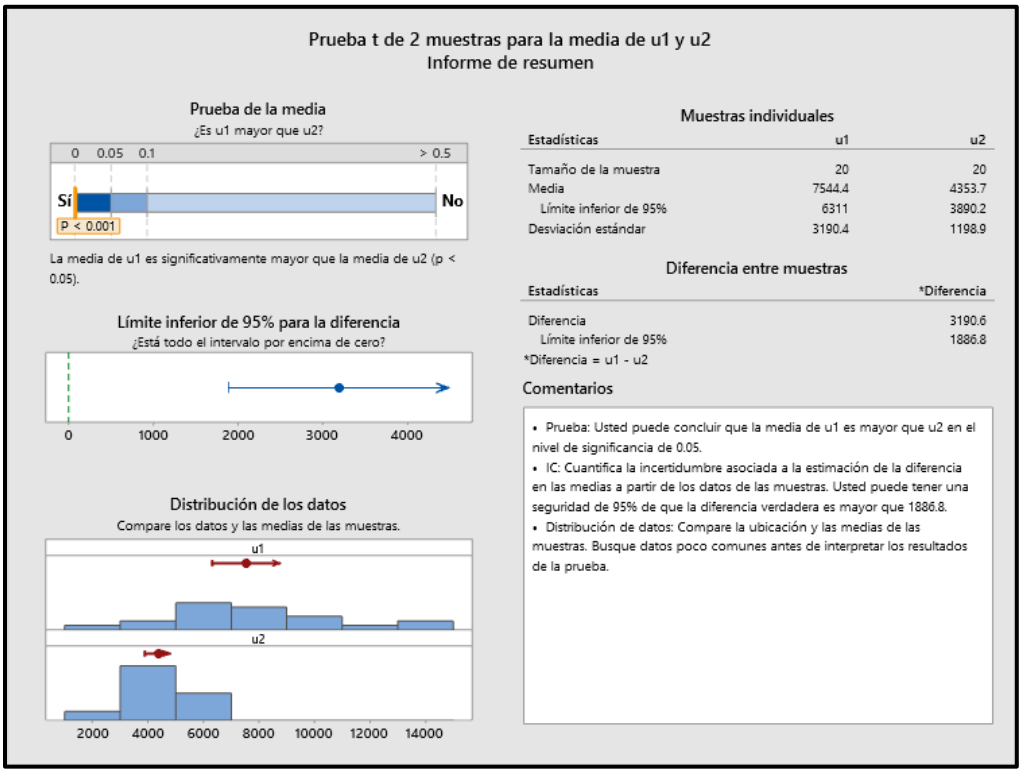
Muestras para la validación de hipotesis en la CPNP Manzano

Datos	Distancia en metros de las rutas									
μ1	13548	7878	12242	3821	8657	7255	5646	2527	13545	9397
	10083	8238	8753	5481	5612	3330	9035	5240	5217	5382
μ2	6531.3	5935.8	6345.8	3612	4163.3	4336.8	3281.5	2331.5	6105.3	5217
	5016.5	4804.5	4141.5	3099.3	3365.5	2879.8	4414.5	4183	3793.5	3516.5

Nota. Los datos son de los resultados de Tabla 30.

Figura 41

Resultados del Minitab para la hipótesis especifica 3



Según la Figura 41 en la sección de comentarios, se rechaza la hipótesis nula por lo tanto se valida $\mu_2 < \mu_1$ o H1. Se concluye que las rutas diseñadas por metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la CPNP Manzano.

Según lo expuesto en las bases teóricas, las metaheurísticas no encuentran un valor óptimo, pero la solución que brinda es el mejor dentro de todos los intentos de búsqueda que se realizó y esto implicaría una supuesta optimización.

5.5.5. Validación de hipótesis especifica 4

H0: Las rutas diseñadas por metaheurísticas no permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la comisaría PNP Ciudad y Campo.

H1: Las rutas diseñadas por metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la comisaría PNP Ciudad y Campo.

Las muestras se ven la Tabla 46 y los resultados del Minitab en la Figura 42.

Tabla 46

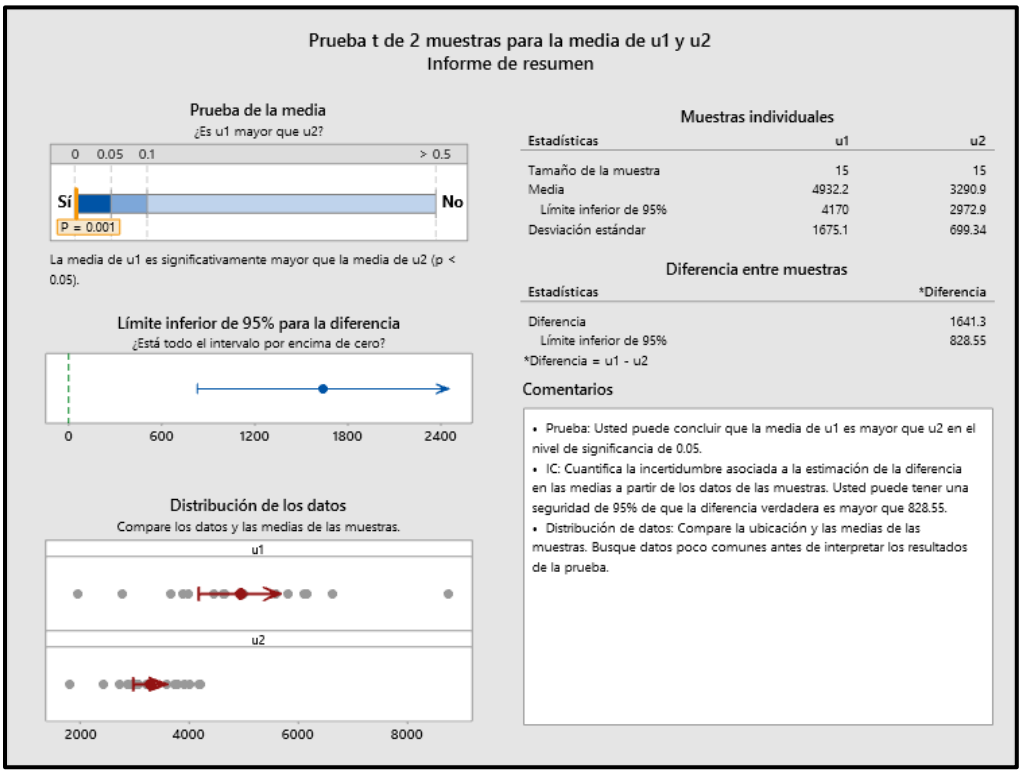
Muestras para la validación de hipotesis en la CPNP Ciudad y Campo

Datos	Distancia en metros de las rutas							
μ1	8766	5002	5577	3987	4622	3662	2754	1947
	6631	5808	4637	3888	6100	6150	4452	
μ2	4181.5	3209.3	3581	2995.5	3053.8	2417	2714.8	1789
	4204.8	3782.5	3737	2849.8	4008	3894.3	2945.3	

Nota. Se usaron datos de la Tabla 37, añadiendo nuevas soluciones con los procedimientos de la Seccion V.4.

Figura 42

Resultados del Minitab para la hipótesis especifica 4



Según la Figura 42 en la sección de comentarios, se rechaza la hipótesis nula por lo tanto se valida $\mu_2 < \mu_1$ o H1. Se concluye que las rutas diseñadas por metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la CPNP Ciudad y Campo.

Según lo expuesto en las bases teóricas, las metaheurísticas no encuentran un valor óptimo, pero la solución que brinda es el mejor dentro de todos los intentos de búsqueda que se realizó y esto implicaría una supuesta optimización.

5.5.6. Validación de hipótesis especifica 5

H0: Las rutas diseñadas por metaheurísticas no permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la comisaría PNP Flor de Amancaes.

H1: Las rutas diseñadas por metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la comisaría PNP Flor de Amancaes.

Las muestras se ven en la Tabla 47 y los resultados con Minitab en la Figura 43.

Tabla 47

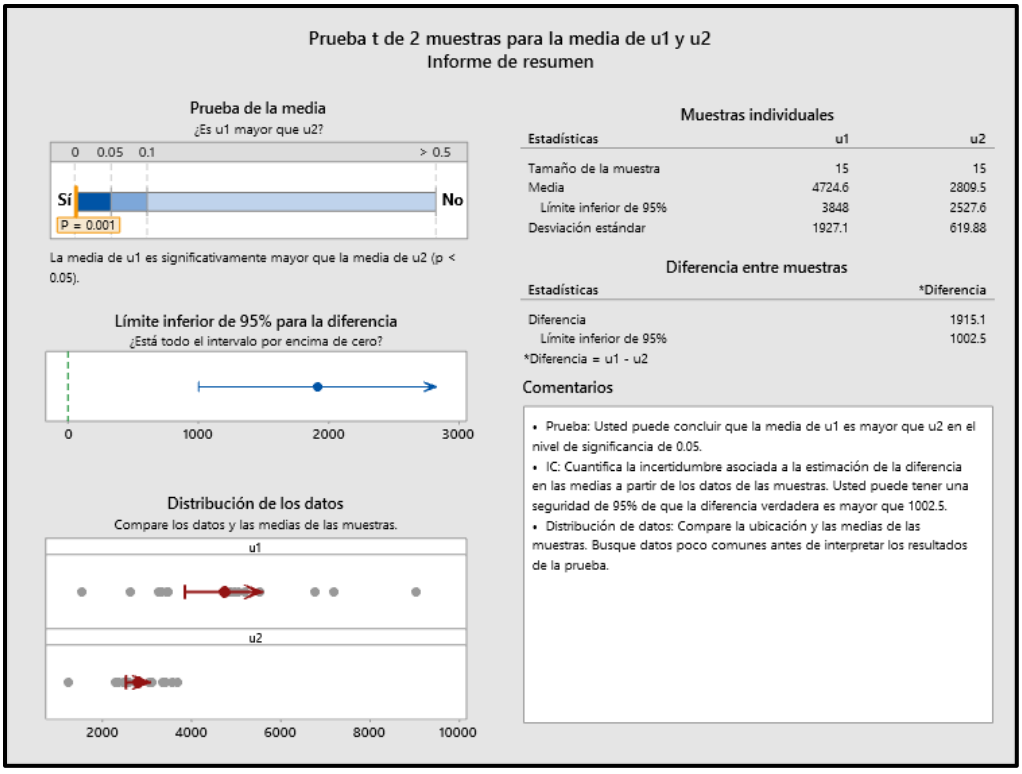
Muestras para la validación de hipotesis en la CPNP Flor de Amancaes

Datos	Distancia en metros de las rutas							
μ1	9038	7184	5182	3448	4933	5509	2614	1517
	3252	4866	4764	3451	6769	3307	5035	
μ2	3544.5	3334.8	3024.5	2592.5	2832.5	3103	2347	1207.8
	2285.3	3673	3057.7	2467.7	3389	2723	2561	

Nota. Se usaron datos de la Tabla 42 y añadiendo nuevas soluciones con los procedimientos de la Seccion V.4

Figura 44

Resultados del Minitab para la hipótesis especifica 5



Según la Figura 44 en la sección de comentarios, se rechaza la hipótesis nula por lo tanto se valida $\mu_2 < \mu_1$ o H1. Se concluye que las rutas diseñadas por metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la CPNP Flor de Amancaes.

Según lo expuesto en las bases teóricas, las metaheurísticas no encuentran un valor óptimo, pero la solución que brinda es el mejor dentro de todos los intentos de búsqueda que se realizó y esto implicaría una supuesta optimización.

5.6. Propuesta de rutas para patrullaje vehicular

Se va a presentar propuesta de rutas para cada comisaria con el uso del aplicativo hecho en esta investigación. Según la elección de los puntos a visitar y el punto de partida, el aplicativo puede ofrecer muchas opciones de rutas.

Se seleccionó todos los puntos de cada sector, un punto de partida al azar y el máximo número de destinos a visitar será de cinco a nueve porque es la decisión más recurrente en las rutinas de patrullaje.

Tabla 48

Propuesta de rutas para CPNP Piedra Liza Sector S1 - 1

Max. Puntos	Distancia (metros)	Ruta
5	2613	PT2 PP4 PC3 PC2 PC5 PP3 PT2
	2322	PT2 PP1 PC5 PP3 PC7 PC8 PT2
	864	PT2 PP2 PC4 PC8 PC6 PC1 PT2
	746	PT2 PP2 PP5 PC4 PC8 PC7 PT2
	737	PT2 PP2 PC8 PC7 PC4 PP5 PT2
	695	PT2 PP5 PC4 PP2 PC6 PC1 PT2
	646	PT2 PP4 PC3 PC1 PC2 PC6 PT2
	553	PT2 PP4 PC2 PC3 PC1 PC6 PT2
6	2503	PT2 PC4 PC5 PP1 PC8 PP2 PC3 PT2
	2326	PT2 PP1 PC5 PP3 PC7 PC8 PC4 PT2
	1675	PT2 PC2 PC1 PC6 PP3 PC7 PP5 PT2
	1406	PT2 PP5 PC4 PC8 PC7 PP3 PP1 PT2
	1374	PT2 PP5 PC4 PC8 PP3 PP1 PC7 PT2
	778	PT2 PP5 PP2 PC6 PC1 PC3 PC2 PT2
	746	PT2 PP4 PC3 PC1 PC2 PC6 PP2 PT2
	653	PT2 PP4 PC2 PC3 PC1 PC6 PP2 PT2
7	2695	PT2 PC2 PC1 PC6 PC8 PC5 PP5 PP2 PT2
	2328	PT2 PP1 PC5 PP3 PC7 PC8 PC4 PP5 PT2
	2318	PT2 PC4 PC8 PC7 PP3 PP1 PC5 PT2
	2285	PT2 PC4 PC8 PP3 PP1 PC5 PC7 PT2
	1584	PT2 PC7 PP3 PP1 PC4 PC3 PP4 PT2
	878	PT2 PP4 PC3 PC1 PC2 PC6 PP2 PP5 PT2
	785	PT2 PP4 PC2 PC3 PC1 PC6 PP2 PP5 PT2
	749	PT2 PC3 PC1 PC2 PP4 PC6 PP2 PT2

Max. Puntos	Distancia (metros)	Ruta
8	2581	PT2 PC5 PP2 PC6 PC1 PC3 PT2
	2340	PT2 PP1 PC5 PP3 PC7 PC8 PC4 PP5 PP2 PT2
	2314	PT2 PC8 PC7 PP3 PP1 PC5 PT2
	2281	PT2 PC8 PP3 PP1 PC5 PC7 PT2
	1654	PT2 PC2 PP4 PC4 PC8 PP3 PP1 PC7 PP5 PT2
	936	PT2 PP4 PC3 PC1 PC2 PC6 PP2 PP5 PC4 PT2
	837	PT2 PP4 PC2 PC3 PC1 PC6 PP2 PC4 PP5 PT2
	649	PT2 PC3 PC1 PC2 PP4 PC6 PT2
	487	PT2 PC3 PC1 PC2 PP4 PT2
9	2599	PT2 PP5 PC4 PC8 PP3 PC5 PC7 PP2 PC1 PC3 PT2
	2392	PT2 PP1 PC5 PP3 PC7 PC8 PC4 PP5 PP2 PC6 PT2
	2278	PT2 PC7 PP3 PP1 PC5 PT2
	2276	PT2 PC7 PC5 PP1 PP3 PT2
	1670	PT2 PC2 PP4 PC6 PP1 PT2
	1118	PT2 PP4 PC3 PC1 PC2 PC6 PP2 PP5 PC4 PC8 PT2
	1014	PT2 PP4 PC2 PC3 PC1 PC6 PC8 PC4 PP5 PP2 PT2
	487	PT2 PC3 PC1 PC2 PP4 PT2

Nota. Los puntos se observan en la Figura 12. PT_i= Puesto Táctico, PC_i= Punto Crítico, PP_i= Punto Protección.

Tabla 49

Propuesta de rutas para CPNP Piedra Liza Sector S2 - 1

Max. Puntos	Distancia (metros)	Ruta
5	1715	PT9 PC16 PC18 PP8 PC14 PC15 PT9
	1702	PT9 PP9 PP8 PC18 PC14 PC17 PT9
	1603	PT9 PC15 PC16 PC17 PC18 PP9 PT9
	1594	PT9 PC15 PC17 PC18 PP9 PC16 PT9
6	1741	PT9 PC17 PC18 PP9 PP8 PC14 PC15 PT9
	1720	PT9 PC15 PC16 PC17 PC18 PP9 PP8 PT9
	1706	PT9 PP9 PP8 PC18 PC14 PC17 PC16 PT9
	1699	PT9 PC15 PC17 PC18 PP8 PP9 PC16 PT9
7	1826	PT9 PP9 PP8 PC18 PC14 PC17 PC16 PC15 PT9
	1759	PT9 PC15 PC16 PC17 PC18 PP9 PP8 PC14 PT9
	1745	PT9 PC15 PC14 PP8 PP9 PC18 PC17 PC16 PT9

Nota. Los puntos se observan en la Figura 14, donde PT_i= Puesto Táctico, PC_i= Punto Crítico, PP_i= Punto Protección.

Tabla 50

Propuesta de rutas para CPNP Rímac Sector S1

Max. Puntos	Distancia (metros)	Ruta									
5	2948	PT2	PP3	PP1	PP4	PC7	PC5	PT2			
	2642	PT2	PP5	PP2	PP3	PP1	PC3	PT2			
	2364	PT2	PP5	PP2	PC4	PC5	PP4	PT2			
	2193	PT2	PP5	PP2	PC5	PP4	PC4	PT2			
	2064	PT2	PC1	PC6	PC5	PP4	PC4	PT2			
	1881	PT2	PP2	PC4	PC6	PC3	PP5	PT2			
	1347	PT2	PC1	PC2	PC8	PC3	PC6	PT2			
	1261	PT2	PC1	PC2	PC6	PC3	PC8	PT2			
6	2974	PT2	PP3	PP1	PP4	PC7	PC5	PP2	PT2		
	2550	PT2	PP5	PP2	PC5	PP4	PP3	PC7	PT2		
	2485	PT2	PP5	PC7	PP3	PP4	PC5	PP2	PT2		
	2457	PT2	PC8	PC3	PC5	PP4	PC7	PC2	PT2		
	2303	PT2	PC6	PC4	PP3	PP2	PP5	PC1	PT2		
	1913	PT2	PC6	PC4	PC3	PP5	PC2	PC8	PT2		
	1546	PT2	PC1	PC2	PC8	PC3	PC6	PC4	PT2		
	1435	PT2	PC1	PC2	PC3	PC6	PC4	PC8	PT2		
7	3142	PT2	PP5	PC5	PP4	PP3	PC7	PP1	PT2		
	3069	PT2	PP3	PP1	PP4	PC7	PC5	PP2	PC4	PT2	
	2705	PT2	PP5	PC7	PP3	PP1	PP4	PC5	PT2		
	2696	PT2	PP5	PC7	PP3	PP1	PC5	PC6	PC2	PT2	
	2140	PT2	PC8	PC3	PC4	PP4	PP2	PC1	PT2		
	1810	PT2	PC6	PC3	PP5	PC2	PC8	PC1	PT2		
	1733	PT2	PC1	PC2	PC8	PC3	PC6	PC4	PP2	PT2	
	1603	PT2	PC1	PC2	PC3	PC6	PP2	PC4	PC8	PT2	
8	3096	PT2	PP3	PP1	PP4	PC7	PC5	PP2	PC4	PC6	PT2
	2841	PT2	PP5	PC7	PP3	PP4	PP1	PT2			
	2826	PT2	PC8	PC2	PC3	PC5	PP4	PP1	PC7	PP5	PT2
	2682	PT2	PP5	PC7	PP3	PP1	PP4	PT2			
	2179	PT2	PC1	PC6	PP2	PP3	PC4	PT2			
	2019	PT2	PC1	PC2	PC8	PC3	PC6	PC4	PP2	PC5	PT2
	1805	PT2	PC1	PC2	PC3	PC6	PP2	PC5	PC4	PC8	PT2
	1362	PT2	PP5	PC3	PC2	PC8	PC1	PT2			

Max. Puntos	Distancia (metros)	Ruta										
9	3106	PT2	PP3	PP1	PP4	PC7	PC5	PP2	PC4	PC6	PC3	PT2
	2750	PT2	PC1	PP5	PC7	PP3	PP1	PP4	PP2	PC3	PC2	PT2
	2681	PT2	PP5	PC7	PP3	PP1	PT2					
	2390	PT2	PC1	PC2	PC8	PC3	PC6	PC4	PP2	PC5	PP4	PT2
	2166	PT2	PC1	PC2	PC3	PC6	PP2	PC5	PP4	PC4	PC8	PT2
	1723	PT2	PC6	PC5	PC4	PC8	PT2					
	1136	PT2	PC8	PC2	PP5	PC1	PT2					

Nota. Los puntos se observan en la Figura 16, donde PT_i= Puesto Táctico, PC_i= Punto Crítico, PP_i= Punto Protección.

Tabla 51

Propuesta de rutas para CPNP Rímac Sector S2

Max. Puntos	Distancia (metros)	Ruta									
5	4820	PT7	PC14	PC10	PP6	PP7	PC11	PT7			
	4809	PT7	PP7	PP6	PC14	PC10	PT7				
	4760	PT7	PP8	PC10	PC12	PC13	PT7				
	4754	PT7	PP7	PC10	PC14	PP6	PT7				
	4509	PT7	PP6	PC14	PP7	PC11	PC9	PT7			
	2918	PT7	PP8	PC13	PC9	PC11	PC12	PT7			
	2886	PT7	PP8	PC9	PC11	PC12	PC13	PT7			
	2720	PT7	PC12	PC9	PC13	PP8	PT7				
6	5065	PT7	PC14	PC10	PP6	PP7	PC11	PC12	PT7		
	3650	PT7	PP8	PC9	PC11	PP6	PC12	PC13	PT7		
	3554	PT7	PP8	PC13	PC9	PC11	PC12	PP7	PT7		
	3309	PT7	PP8	PC9	PC11	PP7	PC12	PC13	PT7		
7	5098	PT7	PC14	PC10	PP6	PP7	PC11	PC12	PC9	PT7	
	4762	PT7	PC9	PP7	PC10	PC14	PP6	PC12	PC13	PT7	
	3975	PT7	PP8	PC13	PC9	PC11	PC12	PP7	PP6	PT7	
	3675	PT7	PP8	PC9	PC11	PP7	PP6	PC12	PC13	PT7	
8	5125	PT7	PC14	PC10	PP6	PP7	PC11	PC12	PC9	PC13	PT7
	4887	PT7	PP8	PC13	PC9	PC11	PC12	PP7	PP6	PC14	PT7
	4802	PT7	PC13	PC9	PC11	PP7	PC10	PC14	PP6	PC12	PT7
	4555	PT7	PP8	PC9	PC11	PP7	PC14	PP6	PC12	PC13	PT7

Max. Puntos	Distancia (metros)	Ruta										
9	5173	PT7	PP8	PC13	PC9	PC11	PC12	PP7	PP6	PC14	PC10	PT7
	5173	PT7	PC14	PC10	PP6	PP7	PC11	PC12	PC9	PC13	PP8	PT7
	4818	PT7	PP8	PC9	PC11	PP7	PC10	PC14	PP6	PC12	PC13	PT7

Nota. Los puntos se observan en la Figura 17, donde PT_i= Puesto Táctico, PC_i=Punto Crítico, PP_i= Punto Protección.

Tabla 52

Propuesta de rutas para CPNP Rímac Sector S3

Max. Puntos	Distancia (metros)	Ruta								
5	6291	PT11	PC26	PC24	PC25	PP11	PP10	PT11		
	6201	PT11	PC23	PP11	PC24	PC26	PP9	PT11		
	6165	PT11	PC19	PP10	PP11	PC25	PC26	PT11		
	6140	PT11	PC19	PP11	PC26	PC25	PP10	PT11		
	6008	PT11	PC20	PC19	PC25	PP10	PC21	PT11		
	2989	PT11	PC21	PC19	PC20	PC22	PC23	PT11		
	2517	PT11	PC16	PC23	PC22	PC20	PC21	PT11		
	2386	PT11	PC16	PC22	PC21	PC20	PC23	PT11		
	1298	PT11	PC27	PC15	PC17	PC16	PC18	PT11		
	1214	PT11	PC17	PC16	PP9	PC18	PC15	PT11		
	1182	PT11	PC15	PC27	PC18	PP9	PC17	PT11		
	1170	PT11	PC27	PC15	PC17	PP9	PC18	PT11		
6	6303	PT11	PC26	PC24	PC25	PP11	PP10	PC19	PT11	
	6261	PT11	PP11	PC25	PC26	PC24	PT11			
	6247	PT11	PP11	PC24	PC26	PC25	PT11			
	6202	PT11	PC15	PC17	PC19	PC26	PC25	PP10	PT11	
	6181	PT11	PP9	PC16	PC24	PP11	PC20	PC23	PT11	
	3873	PT11	PC23	PC22	PC20	PC21	PC19	PP10	PT11	
	3716	PT11	PC23	PC20	PC19	PP10	PC21	PC22	PT11	
	2556	PT11	PC21	PC20	PC22	PC23	PC16	PC17	PT11	
	2505	PT11	PC21	PC22	PC18	PC27	PT11			
	1343	PT11	PC15	PC27	PC18	PP9	PC17	PC16	PT11	
	1298	PT11	PC27	PC15	PC17	PC16	PP9	PC18	PT11	
	889	PT11	PP9	PC18	PC15	PC27	PT11			
7	6358	PT11	PC22	PC21	PC26	PC24	PC20	PC15	PC27	PT11
	6307	PT11	PC26	PC24	PC25	PP11	PP10	PC19	PC21	PT11
	6097	PT11	PC22	PC20	PC21	PC19	PP10	PP11	PC25	PT11
	6017	PT11	PC22	PC21	PP10	PC25	PP11	PC19	PC20	PT11
	6014	PT11	PC17	PC23	PC19	PP11	PC25	PP10	PC16	PT11
	2376	PT11	PC22	PC20	PC23	PC16	PC17	PP9	PC18	PT11
	1907	PT11	PC15	PC27	PC18	PP9	PC17	PC16	PC23	PT11
	1823	PT11	PC27	PC15	PC17	PC23	PC16	PP9	PC18	PT11

Max. Puntos	Distancia (metros)	Ruta										
8	6350	PT11	PC26	PC24	PC25	PP11	PP10	PC19	PC21	PC20	PT11	
	6329	PT11	PC20	PC21	PC19	PP10	PP11	PC25	PC26	PC24	PT11	
	6300	PT11	PC23	PC20	PC19	PP11	PC26	PC25	PC18	PC27	PT11	
	6254	PT11	PC20	PC19	PP11	PC24	PC26	PC25	PP10	PC21	PT11	
	6238	PT11	PP9	PC16	PC22	PC24	PP10	PC21	PC17	PC15	PT11	
	2069	PT11	PC15	PC27	PC18	PP9	PC17	PC16	PC23	PC22	PT11	
	2048	PT11	PC23	PC22	PC16	PC17	PP9	PC18	PC15	PC27	PT11	
	1965	PT11	PC27	PC15	PC17	PC23	PC22	PC16	PP9	PC18	PT11	
9	6374	PT11	PC26	PC24	PC25	PP11	PP10	PC19	PC21	PC20	PC22	PT11
	6286	PT11	PC21	PC19	PP10	PP11	PC25	PC26	PC24	PT11		
	6266	PT11	PC23	PC24	PC26	PC25	PC22	PC16	PC18	PT11		
	6253	PT11	PC21	PP10	PC25	PC26	PC24	PP11	PC19	PT11		
	4271	PT11	PC27	PC15	PC17	PC20	PC19	PP11	PP10	PC21	PP9	PT11
	2637	PT11	PC15	PC27	PC18	PP9	PC17	PC16	PC23	PC22	PC20	PT11
	2441	PT11	PC27	PC15	PC17	PC23	PC20	PC22	PC16	PP9	PC18	PT11
	1906	PT11	PC23	PC16	PC17	PP9	PC18	PC15	PC27	PT11		

Nota. Los puntos se observan en la Figura 18, donde PT_i= Puesto Táctico, PC_i=Punto Crítico, PP_i= Punto Protección.

Tabla 53

Propuesta de rutas para CPNP Rímac Sector S4

Max. Puntos	Distancia (metros)	Ruta								
5	2963	PT19	PC31	PC34	PC28	PC33	PC32	PT19		
	1939	PT19	PC33	PC28	PC29	PC34	PC30	PT19		
	1360	PT19	PC32	PC30	PC34	PC29	PC28	PT19		
	1353	PT19	PC32	PC30	PC28	PC29	PC34	PT19		
6	3093	PT19	PC32	PC33	PC28	PC29	PC34	PC31	PT19	
	1939	PT19	PC32	PC30	PC34	PC29	PC28	PC33	PT19	
7	3148	PT19	PC32	PC30	PC34	PC29	PC28	PC33	PC31	PT19
	3131	PT19	PC33	PC28	PC29	PC34	PC30	PC32	PC31	PT19
	3100	PT19	PC31	PC34	PC29	PC28	PC33	PC30	PC32	PT19

Nota. Los puntos se observan en la Figura 19, donde PT_i= Puesto Táctico, PC_i=Punto Crítico, PP_i= Punto Protección.

Tabla 54

Propuesta de rutas para CPNP Rímac Sector S5

Max. Puntos	Distancia (metros)	Ruta								
5	3701	PT21	PP17	PC36	PP16	PC35	PC37	PT21		
	3665	PT21	PC37	PP17	PC36	PC35	PC39	PT21		
	2749	PT21	PC39	PC38	PC37	PC35	PP16	PT21		
	2707	PT21	PC39	PC38	PP16	PC35	PC37	PT21		
6	3750	PT21	PP17	PC36	PP16	PC35	PC37	PC38	PT21	
	3707	PT21	PC37	PC35	PC36	PP16	PC38	PC39	PT21	
	3137	PT21	PC39	PC38	PC37	PC35	PP16	PP17	PT21	
	3096	PT21	PC39	PC38	PP16	PP17	PC35	PC37	PT21	
7	3904	PT21	PC39	PC38	PC37	PC35	PP16	PP17	PC36	PT21
	3762	PT21	PP17	PC36	PP16	PC35	PC37	PC38	PC39	PT21
	3721	PT21	PC39	PC38	PP16	PC36	PP17	PC35	PC37	PT21

Nota. Los puntos se observan en la Figura 20, donde PT_i= Puesto Táctico, PC_i=Punto Crítico, PP_i= Punto Protección.

Tabla 55

Propuesta de rutas para CPNP Manzano Sector S2

Max. Puntos	Distancia (metros)	Ruta							
5	6011	PT3	PP3	PC8	PC11	PC10	PC6	PT3	
	5731	PT3	PP3	PC11	PC8	PC12	PC6	PT3	
	5731	PT3	PP3	PC11	PC8	PC6	PC10	PT3	
	3561	PT3	PP3	PC11	PC8	PC7	PC9	PT3	
	3144	PT3	PC7	PC9	PC5	PC10	PP4	PT3	
	3032	PT3	PC6	PC5	PC10	PP4	PC12	PT3	
	1442	PT3	PP2	PC9	PC7	PC12	PP4	PT3	
	1429	PT3	PP2	PC7	PC9	PC12	PP4	PT3	
6	6339	PT3	PP3	PC8	PC11	PC6	PC5	PT3	
	6060	PT3	PC9	PC11	PC8	PC12	PC5	PT3	
	6048	PT3	PP3	PC11	PC8	PC5	PC6	PT3	
	4202	PT3	PC6	PC5	PC10	PP4	PC12	PC8	PT3
	3673	PT3	PC7	PP3	PP2	PP4	PC10	PC6	PT3
	3603	PT3	PP3	PC11	PC7	PC9	PP2	PT3	
	2684	PT3	PP2	PC9	PC7	PC12	PP4	PC10	PT3
	2544	PT3	PP2	PC7	PC9	PC12	PC10	PP4	PT3

Max. Puntos	Distancia (metros)	Ruta										
7	6328	PT3	PP3	PC8	PC11	PC5	PT3					
	6283	PT3	PC6	PC5	PC10	PP4	PC12	PC8	PC11	PT3		
	6039	PT3	PP3	PC11	PC8	PC5	PT3					
	5788	PT3	PC10	PC6	PC12	PC8	PC11	PC7	PP2	PT3		
	4003	PT3	PC5	PP4	PP3	PC9	PT3					
	3143	PT3	PP2	PC9	PC7	PC12	PP4	PC10	PC6	PT3		
	2975	PT3	PP2	PC7	PC9	PC12	PC6	PC10	PP4	PT3		
	1561	PT3	PP3	PC7	PC9	PP2	PT3					
8	6287	PT3	PC6	PC5	PC10	PP4	PC12	PC8	PC11	PP3	PT3	
	6129	PT3	PC9	PP3	PC11	PC8	PC12	PC5	PC6	PP4	PT3	
	3484	PT3	PP2	PC9	PC7	PC12	PP4	PC10	PC6	PC5	PT3	
	3300	PT3	PP2	PC7	PC9	PC12	PC5	PC6	PC10	PP4	PT3	
9	6310	PT3	PC6	PC5	PC10	PP4	PC12	PC8	PC11	PP3	PC7	PT3
	6130	PT3	PC7	PP3	PC11	PC8	PC12	PC5	PC6	PC10	PP4	PT3
	4920	PT3	PP2	PC9	PC7	PC12	PP4	PC10	PC6	PC5	PC8	PT3
	4122	PT3	PP2	PC9	PC7	PC8	PC12	PC5	PC6	PC10	PP4	PT3

Nota. Los puntos se observan en la Figura 23, donde PT_i= Puesto Táctico, PC_i=Punto Crítico, PP_i= Punto Protección.

Tabla 56

Propuesta de rutas para CPNP Manzano Sector S3

Max. Puntos	Distancia (metros)	Ruta							
5	3888	PT7	PC19	PP6	PP5	PC13	PC16	PT7	
	2966	PT7	PC15	PP5	PC17	PC13	PC16	PT7	
	2897	PT7	PC17	PC16	PC13	PP5	PP7	PT7	
	2892	PT7	PC15	PC17	PC16	PC13	PP5	PT7	
	2377	PT7	PC14	PP7	PC17	PC18	PC15	PT7	
	1659	PT7	PC14	PP6	PC18	PP8	PP7	PT7	
	1542	PT7	PP8	PC18	PC15	PC14	PP6	PT7	
	1499	PT7	PC14	PP6	PP7	PP8	PC18	PT7	
6	3952	PT7	PC15	PC17	PC13	PC16	PC19	PT7	
	3889	PT7	PC15	PC17	PC16	PC13	PC19	PT7	
	3852	PT7	PC19	PC14	PP7	PC13	PC16	PC18	PT7
	2963	PT7	PC17	PC16	PC13	PP5	PP7	PP8	PT7
	2443	PT7	PC15	PP8	PC17	PP5	PP6	PT7	
	2257	PT7	PC18	PC15	PC14	PP6	PC19	PT7	
	2234	PT7	PC14	PP6	PC18	PP8	PP7	PP5	PT7
	2009	PT7	PC14	PP6	PP7	PP5	PP8	PC18	PT7

Max. Puntos	Distancia (metros)	Ruta										
7	3896	PT7	PC15	PC13	PC16	PC19	PT7					
	3849	PT7	PC15	PC16	PC13	PC19	PT7					
	3793	PT7	PC19	PC17	PC16	PC15	PT7					
	2975	PT7	PC17	PC16	PC13	PP5	PP7	PP8	PC18	PT7		
	2648	PT7	PC14	PP6	PC18	PP8	PP7	PP5	PC17	PT7		
	2457	PT7	PC14	PP6	PP7	PC13	PP5	PP8	PC18	PT7		
	2422	PT7	PC14	PP6	PP7	PP5	PC17	PP8	PC18	PT7		
	2188	PT7	PP6	PC14	PC15	PC19	PT7					
8	3016	PT7	PC17	PC16	PC13	PP5	PP7	PP8	PC18	PC15	PT7	
	2777	PT7	PC14	PP6	PC18	PP8	PP7	PP5	PC17	PC13	PT7	
	2725	PT7	PC15	PC18	PC17	PC13	PP7	PP8	PP6	PC14	PT7	
	2539	PT7	PC14	PP6	PP7	PP5	PC13	PC17	PP8	PC18	PT7	
9	3425	PT7	PC19	PC14	PP7	PC13	PC17	PP5	PP8	PC18	PC15	PT7
	3341	PT7	PC14	PP6	PC18	PP8	PP7	PP5	PC17	PC13	PC16	PT7
	3316	PT7	PC17	PC16	PC13	PP5	PP7	PP8	PC18	PC15	PC14	PT7
	3058	PT7	PC14	PP6	PP7	PP5	PC13	PC16	PC17	PP8	PC18	PT7

Nota. Los puntos se observan en la Figura 24, donde PT_i= Puesto Táctico, PC_i=Punto Crítico, PP_i= Punto Protección.

Tabla 57

Propuesta de rutas para CPNP Manzano Sector S4

Max. Puntos	Distancia (metros)	Ruta							
5	3840	PT9	PC24	PC23	PC20	PC26	PP9	PT9	
	3815	PT9	PC21	PC20	PC26	PC24	PP11	PT9	
	3168	PT9	PC22	PC25	PC27	PC28	PP9	PT9	
	2254	PT9	PC28	PP10	PC22	PC27	PC25	PT9	
	2099	PT9	PC27	PC25	PC22	PP10	PC28	PT9	
	1991	PT9	PC28	PC27	PC25	PC22	PP10	PT9	
	1824	PT9	PC21	PP9	PP11	PC26	PC20	PT9	
	1720	PT9	PC21	PC20	PC26	PP9	PP11	PT9	
6	4875	PT9	PP10	PC27	PC22	PC20	PC24	PP11	PT9
	4569	PT9	PC28	PC25	PC21	PC26	PC23	PP9	PT9
	4006	PT9	PP9	PP11	PC26	PC20	PC23	PC24	PT9
	3950	PT9	PC24	PC23	PC20	PC26	PP9	PP11	PT9
	3834	PT9	PP9	PP11	PC24	PC23	PC20	PC26	PT9
	2568	PT9	PC28	PP10	PC22	PC27	PC25	PC21	PT9
	2496	PT9	PC27	PC25	PC22	PP10	PC28	PC21	PT9
	2318	PT9	PC28	PC27	PC25	PC22	PP10	PC21	PT9

Max. Puntos	Distancia (metros)	Ruta										
7	5548	PT9	PC26	PC24	PP11	PC28	PC27	PC25	PC22	PT9		
	4061	PT9	PC24	PC23	PC20	PC26	PP9	PP11	PC21	PT9		
	3940	PT9	PP11	PC26	PC20	PC23	PC24	PT9				
	3768	PT9	PP11	PC24	PC23	PC20	PC26	PT9				
	3515	PT9	PC28	PP10	PC22	PC27	PC25	PC21	PP9	PT9		
	3255	PT9	PP10	PC22	PC25	PC27	PC28	PP9	PC21	PT9		
	2999	PT9	PP9	PC23	PC20	PC21	PP10	PT9				
	2099	PT9	PC27	PC25	PC22	PP10	PC28	PT9				
8	4927	PT9	PC24	PC23	PC20	PC26	PP9	PP11	PC21	PC22	PT9	
	4860	PT9	PP10	PC27	PC28	PP11	PC24	PC23	PC20	PC26	PT9	
	3767	PT9	PC26	PC20	PC23	PC24	PT9					
	3758	PT9	PC26	PC24	PC23	PC20	PT9					
	3625	PT9	PC28	PP10	PC22	PC27	PC25	PC21	PP9	PP11	PT9	
	3345	PT9	PP10	PC22	PC25	PC27	PC28	PP11	PP9	PC21	PT9	
	3076	PT9	PC25	PC22	PC21	PP9	PT9					
	2083	PT9	PC27	PC25	PP10	PC28	PT9					
9	5788	PT9	PC24	PC23	PC20	PC26	PP9	PP11	PC21	PC22	PC25	PT9
	5673	PT9	PP9	PC24	PC23	PC20	PP10	PC22	PC25	PC27	PC28	PT9
	3876	PT9	PC28	PP10	PC22	PC27	PC25	PC21	PP9	PP11	PC26	PT9
	3539	PT9	PC28	PC27	PC25	PC22	PP10	PC21	PC26	PP9	PP11	PT9

Nota. Los puntos se observan en la Figura 25, donde PT_i= Puesto Táctico, PC_i=Punto Crítico, PP_i= Punto Protección.

Tabla 58

Propuesta de rutas para CPNP Manzano Sector S5

Max. Puntos	Distancia (metros)	Ruta						
5	3746	PT15	PC31	PC34	PC30	PP12	PC29	PT15
	3574	PT15	PC31	PC34	PC30	PC29	PP12	PT15
	3171	PT15	PC32	PC29	PC30	PP12	PP13	PT15
	3139	PT15	PC34	PC32	PC29	PP12	PC33	PT15
	2796	PT15	PC31	PC35	PP13	PC30	PC36	PT15
	1582	PT15	PP14	PC36	PC35	PC33	PP13	PT15
	1509	PT15	PP14	PC35	PC33	PP13	PC36	PT15
	1472	PT15	PC34	PC33	PC35	PC36	PP14	PT15

Max. Puntos	Distancia (metros)	Ruta										
6	3915	PT15	PC31	PP12	PC30	PC29	PC32	PT15				
	3880	PT15	PC31	PP12	PC29	PC32	PC30	PT15				
	3861	PT15	PC31	PC35	PP13	PP12	PC32	PT15				
	3300	PT15	PC32	PC29	PC30	PP12	PP13	PC34	PT15			
	2825	PT15	PP14	PC33	PC30	PC29	PC34	PC36	PT15			
	1711	PT15	PP14	PC36	PC35	PC33	PP13	PC34	PT15			
	1634	PT15	PC35	PC33	PC36	PP14	PC31	PT15				
	1628	PT15	PP14	PC35	PC33	PP13	PC34	PC36	PT15			
7	3882	PT15	PC31	PP14	PP12	PC29	PC32	PC30	PC36	PT15		
	3860	PT15	PC31	PP12	PC29	PC32	PT15					
	3858	PT15	PC31	PC32	PC29	PP12	PT15					
	3446	PT15	PC32	PC29	PC30	PP12	PP13	PC34	PC33	PT15		
	2543	PT15	PP14	PC36	PC35	PC33	PP13	PC34	PC30	PT15		
	2134	PT15	PP14	PC35	PC33	PP13	PC30	PC34	PC36	PT15		
	1592	PT15	PC35	PC33	PP13	PC34	PT15					
	1368	PT15	PC36	PC35	PP14	PC31	PT15					
8	3845	PT15	PC31	PP14	PC35	PC33	PP13	PC30	PC32	PC34	PT15	
	3449	PT15	PC32	PC29	PC30	PP12	PP13	PC34	PC33	PC35	PT15	
	2667	PT15	PP14	PC36	PC35	PC33	PP13	PC34	PC30	PP12	PT15	
	2225	PT15	PP14	PC35	PC33	PP13	PP12	PC30	PC34	PC36	PT15	
9	3886	PT15	PC36	PC30	PC29	PC32	PP12	PP13	PC35	PP14	PC31	PT15
	3512	PT15	PP14	PC36	PC35	PC33	PP13	PC34	PC30	PP12	PC29	PT15
	3511	PT15	PC32	PC29	PC30	PP12	PP13	PC34	PC33	PC35	PC36	PT15
	2898	PT15	PP14	PC35	PC33	PP13	PP12	PC29	PC30	PC34	PC36	PT15

Nota. Los puntos se observan en la Figura 26, donde PT_i= Puesto Táctico, PC_i=Punto Crítico, PP_i= Punto Protección.

Tabla 59

Propuesta de rutas para CPNP Manzano Sector S6

Max. Puntos	Distancia (metros)	Ruta
5	3190	PT18 PP15 PC40 PC42 PC43 PP16 PT18
	3061	PT18 PC44 PP17 PP18 PC41 PP15 PT18
	3031	PT18 PP15 PC42 PC43 PC40 PP16 PT18
	2978	PT18 PC38 PP16 PC40 PC43 PC42 PT18
	2767	PT18 PP17 PC39 PP16 PC37 PC38 PT18
	2227	PT18 PC43 PC42 PP15 PC40 PP18 PT18
	2132	PT18 PC41 PP18 PC38 PC44 PC37 PT18
	1673	PT18 PC41 PP18 PC37 PC38 PC44 PT18
6	4124	PT18 PP15 PC40 PC42 PC43 PP17 PC39 PT18
	3950	PT18 PP15 PC42 PC43 PC40 PP17 PC39 PT18
	3598	PT18 PP17 PC37 PP18 PC40 PC43 PC42 PT18
	3290	PT18 PP15 PC41 PP16 PC39 PC38 PC44 PT18
	2832	PT18 PP17 PC39 PP16 PC37 PC38 PC44 PT18
	2407	PT18 PC41 PP18 PC38 PC44 PC37 PP16 PT18
	2250	PT18 PC43 PC42 PP15 PC40 PP18 PC41 PT18
	1859	PT18 PC41 PP18 PC37 PP16 PC38 PC44 PT18
7	4122	PT18 PP15 PC40 PC42 PC43 PC39 PT18
	3950	PT18 PP15 PC42 PC43 PC40 PC39 PT18
	3648	PT18 PC41 PC38 PC39 PP18 PC40 PP15 PC42 PT18
	3610	PT18 PC43 PC37 PP17 PP16 PC44 PT18
	3184	PT18 PP17 PC39 PP16 PC37 PC38 PC44 PP18 PT18
	3056	PT18 PC41 PP18 PC38 PC44 PC37 PP16 PP17 PT18
	2447	PT18 PC41 PP18 PC37 PP17 PP16 PC38 PC44 PT18
	2047	PT18 PC43 PC42 PP15 PC40 PC41 PT18
8	3978	PT18 PC44 PC38 PC39 PC37 PP18 PC41 PP15 PC43 PT18
	3348	PT18 PC41 PP18 PC38 PC44 PC37 PP16 PP17 PC39 PT18
	3296	PT18 PC42 PC40 PP17 PP16 PT18
	3207	PT18 PP17 PC39 PP16 PC37 PC38 PC44 PP18 PC41 PT18
	2736	PT18 PC41 PP18 PC37 PP17 PC39 PP16 PC38 PC44 PT18
	1555	PT18 PP15 PC40 PC42 PC43 PT18
	1512	PT18 PC43 PC42 PP15 PC40 PT18
	1490	PT18 PP15 PC42 PC43 PC40 PT18

Max. Puntos	Distancia (metros)	Ruta										
9	4038	PT18	PC41	PP18	PC38	PC44	PC37	PP16	PP17	PC39	PC40	PT18
	4025	PT18	PC44	PC38	PP16	PC39	PP18	PC40	PC43	PC42	PP15	PT18
	3910	PT18	PP17	PC39	PP16	PC37	PC38	PC44	PP18	PC41	PC40	PT18
	3439	PT18	PC44	PC38	PP16	PC39	PP17	PC37	PP18	PC41	PC40	PT18

Nota. Los puntos se observan en la Figura 27, donde PT_i= Puesto Táctico, PC_i=Punto Crítico, PP_i= Punto Protección.

Tabla 60

Propuesta de rutas para CPNP Ciudad y Campo Sector S1 – 2

Max. Puntos	Distancia (metros)	Ruta										
5	3904	PT4	PP5	PP1	PC2	PP3	PT4					
	2866	PT4	PC4	PC3	PP5	PP1	PP4	PT4				
	2770	PT4	PC3	PP1	PP5	PC4	PP4	PT4				
	2552	PT4	PC3	PC4	PP4	PP2	PC5	PT4				
	2228	PT4	PP5	PP1	PC3	PC4	PT4					
	2218	PT4	PP2	PC2	PP3	PC5	PP4	PT4				
	2002	PT4	PC5	PP3	PC2	PP2	PT4					
6	4128	PT4	PP2	PC2	PP3	PC5	PP4	PP1	PT4			
	3933	PT4	PC2	PP4	PC4	PP5	PP1	PC3	PT4			
	3390	PT4	PC4	PC3	PP5	PP1	PP4	PC5	PT4			
	3294	PT4	PC3	PP1	PP5	PC4	PP4	PC5	PT4			
7	4129	PT4	PP2	PC2	PP3	PC5	PP4	PP1	PP5	PT4		
	3966	PT4	PC3	PP1	PP5	PC4	PP4	PP2	PC2	PT4		
	3679	PT4	PC4	PC3	PP5	PP1	PP4	PC5	PP3	PT4		
	3583	PT4	PC3	PP1	PP5	PC4	PP4	PC5	PP3	PT4		
8	4198	PT4	PP2	PC2	PP3	PC5	PP4	PP1	PP5	PC3	PT4	
	4193	PT4	PC4	PC3	PP5	PP1	PP4	PC5	PP3	PC2	PT4	
	4116	PT4	PC4	PC3	PP5	PP1	PP4	PP2	PC2	PP3	PT4	
	4022	PT4	PC3	PP1	PP5	PC4	PP4	PC2	PP3	PC5	PT4	
9	4298	PT4	PC4	PC3	PP5	PP1	PP4	PC5	PP3	PC2	PP2	PT4
	4298	PT4	PP2	PC2	PP3	PC5	PP4	PP1	PP5	PC3	PC4	PT4
	4055	PT4	PC5	PP3	PC2	PP2	PP4	PC4	PP5	PP1	PC3	PT4
	4055	PT4	PC3	PP1	PP5	PC4	PP4	PP2	PC2	PP3	PC5	PT4

Nota. Los puntos se observan en la Figura 29, donde PT_i= Puesto Táctico, PC_i=Punto Crítico, PP_i= Punto Protección.

Tabla 61

Propuesta de rutas para CPNP Ciudad y Campo Sector S2 – 1

Max. Puntos	Distancia (metros)	Ruta										
5	2586	PT6	PC9	PC7	PP9	PC10	PT6					
	2567	PT6	PP9	PC7	PC8	PC10	PP7	PT6				
	2509	PT6	PP7	PC8	PC7	PP9	PT6					
	2392	PT6	PP7	PP9	PC7	PC8	PT6					
	2130	PT6	PP6	PP8	PC9	PC6	PC10	PT6				
	1932	PT6	PC6	PP8	PC8	PP7	PP6	PT6				
	1640	PT6	PC6	PC9	PP8	PC10	PP6	PT6				
	1177	PT6	PP8	PC9	PC6	PP6	PT6					
6	2743	PT6	PP9	PC7	PC8	PC10	PP7	PP8	PT6			
	2416	PT6	PP6	PC10	PC8	PC7	PP9	PP7	PT6			
	2348	PT6	PP6	PP8	PC9	PC6	PC10	PC8	PT6			
	1823	PT6	PC6	PC9	PP8	PC8	PC10	PP6	PT6			
7	2868	PT6	PP9	PC7	PC8	PC10	PP7	PP8	PC9	PT6		
	2741	PT6	PP6	PC10	PP9	PC7	PC8	PC9	PC6	PT6		
	2639	PT6	PP6	PP8	PC9	PC6	PC10	PC8	PC7	PT6		
	2107	PT6	PC6	PC9	PP8	PC7	PC8	PC10	PP6	PT6		
8	3008	PT6	PP9	PC7	PC8	PC10	PP7	PP8	PC9	PC6	PT6	
	2820	PT6	PP6	PC10	PP7	PP9	PC8	PP8	PC9	PC6	PT6	
	2716	PT6	PP6	PP8	PC9	PC6	PC10	PC8	PC7	PP7	PT6	
	2237	PT6	PC6	PC9	PP8	PC10	PC8	PC7	PP7	PP6	PT6	
9	3743	PT6	PP6	PP8	PC9	PC6	PC10	PC8	PC7	PP7	PP9	PT6
	3284	PT6	PP9	PC7	PC8	PC10	PP7	PP8	PC9	PC6	PP6	PT6
	2825	PT6	PP6	PP7	PP9	PC7	PC8	PC10	PP8	PC9	PC6	PT6
	2825	PT6	PC6	PC9	PP8	PC10	PC8	PC7	PP9	PP7	PP6	PT6

Nota. Los puntos se observan en la Figura 30, donde PT_i= Puesto Táctico, PC_i=Punto Crítico, PP_i= Punto Protección.

Tabla 62

Propuesta de rutas para CPNP Ciudad y Campo Sector S3 – 1

Max. Puntos	Distancia (metros)	Ruta
5	3826	PT13 PC14 PP10 PP13 PC15 PP14 PT13
	3826	PT13 PP14 PC15 PP13 PP10 PC14 PT13
	3708	PT13 PC14 PP10 PC15 PP14 PP13 PT13
	3614	PT13 PC17 PP14 PC15 PC16 PC13 PT13
	2604	PT13 PP11 PP13 PP10 PC14 PP12 PT13
	1801	PT13 PP11 PC16 PC17 PC13 PP12 PT13
	1771	PT13 PP12 PC13 PC16 PC17 PP11 PT13
	1529	PT13 PP12 PC17 PP11 PC16 PC13 PT13
6	3879	PT13 PP14 PC15 PP13 PP10 PC14 PP11 PT13
	3754	PT13 PP10 PP13 PC15 PP14 PT13
	3679	PT13 PC13 PC16 PP13 PP14 PC15 PC17 PT13
	3636	PT13 PP10 PC15 PP14 PP13 PT13
	2369	PT13 PP11 PP10 PC14 PP12 PT13
	2078	PT13 PP12 PC13 PC16 PC17 PP11 PC14 PT13
	1804	PT13 PP12 PC17 PC14 PP11 PC16 PC13 PT13
	1148	PT13 PC17 PC16 PC13 PP12 PT13
7	3922	PT13 PP14 PC15 PP13 PP10 PC14 PP11 PC16 PT13
	3652	PT13 PP11 PP10 PC15 PP14 PP13 PC16 PC13 PT13
	2746	PT13 PP12 PC13 PC16 PC17 PP11 PC14 PP10 PT13
	2422	PT13 PP12 PC17 PC14 PP10 PP11 PC16 PC13 PT13
8	4155	PT13 PP14 PC15 PP13 PP10 PC14 PP11 PC16 PC17 PT13
	3805	PT13 PC13 PC16 PP13 PC15 PP14 PC14 PP11 PP12 PT13
	2956	PT13 PP12 PC13 PC16 PC17 PP11 PC14 PP10 PP13 PT13
	2657	PT13 PP12 PC17 PC14 PP10 PP13 PP11 PC16 PC13 PT13
9	4205	PT13 PP14 PC15 PP13 PP10 PC14 PP11 PC16 PC17 PC13 PT13
	3843	PT13 PP12 PC14 PP10 PC15 PP14 PP13 PP11 PC16 PC13 PT13
	3291	PT13 PP12 PC13 PC16 PC17 PP11 PC14 PP10 PP13 PC15 PT13
	2900	PT13 PP12 PC17 PC14 PP10 PC15 PP13 PP11 PC16 PC13 PT13

Nota. Los puntos se observan en la Figura 32, donde PT_i= Puesto Táctico, PC_i=Punto Crítico, PP_i= Punto Protección.

Tabla 63

Propuesta de rutas para CPNP Flor de Amancaes Sector S1 – 2

Max. Puntos	Distancia (metros)	Ruta										
5	3142	PT6	PC10	PC6	PC8	PC4	PP1	PT6				
	2537	PT6	PP3	PP2	PC12	PC7	PC5	PT6				
	2328	PT6	PC10	PC6	PP2	PC8	PC11	PT6				
	1710	PT6	PP3	PC11	PC9	PC8	PP2	PT6				
	1608	PT6	PP3	PC11	PP2	PC8	PC9	PT6				
	1264	PT6	PP1	PC5	PC12	PC7	PC4	PT6				
	1255	PT6	PC7	PC4	PC5	PC12	PP1	PT6				
	1066	PT6	PP1	PC5	PC4	PC7	PC12	PT6				
6	3176	PT6	PP1	PC4	PC12	PC6	PC10	PC11	PT6			
	2535	PT6	PC9	PC8	PP2	PP3	PC5	PC7	PT6			
	2397	PT6	PC9	PC11	PC8	PP2	PC10	PC6	PT6			
	2374	PT6	PC10	PC6	PP2	PC8	PC11	PC9	PT6			
	2292	PT6	PC9	PC8	PP2	PC6	PC10	PC11	PT6			
	1750	PT6	PP1	PC5	PC12	PC7	PC4	PP3	PT6			
	1743	PT6	PC7	PC4	PC5	PC12	PP1	PP3	PT6			
	1529	PT6	PP1	PC5	PC4	PC7	PC12	PP3	PT6			
7	2649	PT6	PP1	PC12	PC9	PC6	PC10	PT6				
	2572	PT6	PP3	PC11	PP2	PC8	PC7	PC4	PC5	PT6		
	2410	PT6	PC10	PC6	PP2	PC8	PC11	PC9	PP3	PT6		
	2315	PT6	PC9	PC8	PP2	PC10	PC6	PT6				
	2292	PT6	PC9	PC8	PP2	PC6	PC10	PT6				
	2016	PT6	PP1	PC5	PC12	PC7	PC4	PP3	PC11	PT6		
	1778	PT6	PP1	PC5	PC4	PC7	PC12	PC11	PP3	PT6		
	1255	PT6	PC7	PC4	PC5	PC12	PP1	PT6				
8	3260	PT6	PC4	PC7	PC12	PP2	PC6	PC10	PC11	PP3	PT6	
	2779	PT6	PC10	PC6	PP2	PC8	PC11	PC9	PP3	PC12	PT6	
	2313	PT6	PC8	PP2	PC10	PC6	PT6					
	2290	PT6	PC8	PP2	PC6	PC10	PT6					
	2062	PT6	PP1	PC5	PC12	PC7	PC4	PP3	PC11	PC9	PT6	
	1804	PT6	PP1	PC5	PC4	PC7	PC12	PC9	PC11	PP3	PT6	
	1447	PT6	PC5	PP1	PC8	PC9	PT6					
	987	PT6	PC7	PC4	PC5	PP1	PT6					
9	3372	PT6	PC10	PC6	PP2	PC8	PC11	PC9	PP3	PC12	PC7	PT6
	3236	PT6	PP3	PC11	PC6	PP2	PC8	PC12	PC7	PC5	PP1	PT6
	2434	PT6	PP1	PC5	PC12	PC7	PC4	PP3	PC11	PC9	PC8	PT6
	2110	PT6	PP1	PC5	PC4	PC7	PC12	PC9	PC8	PC11	PP3	PT6

Nota. Los puntos se observan en la Figura 35, donde PT_i= Puesto Táctico, PC_i=Punto Crítico, PP_i= Punto Protección.

Tabla 64

Propuesta de rutas para CPNP Flor de Amancaes Sector S2 – 2

Max. Puntos	Distancia	Ruta										
5	2784	PT17	PP5	PP4	PC24	PC22	PC16	PT17				
	2567	PT17	PC18	PC22	PC24	PP4	PC16	PT17				
	2000	PT17	PC18	PC22	PC24	PP5	PC16	PT17				
	1527	PT17	PC20	PC18	PC21	PC23	PC19	PT17				
	1221	PT17	PC17	PC19	PC20	PC21	PC23	PT17				
6	2881	PT17	PP5	PP4	PC24	PC22	PC16	PC18	PT17			
	2762	PT17	PC16	PC22	PC24	PP5	PP4	PT17				
	2744	PT17	PC17	PP4	PP5	PC22	PC18	PC21	PT17			
	2593	PT17	PC16	PP4	PP5	PC24	PC22	PT17				
	2109	PT17	PC23	PC20	PC24	PC16	PC19	PT17				
	1697	PT17	PC17	PC19	PC20	PC21	PC23	PC18	PT17			
	1412	PT17	PC21	PC20	PC23	PC19	PC17	PT17				
	1380	PT17	PC17	PC19	PC18	PC20	PC21	PC23	PT17			
7	2974	PT17	PP5	PP4	PC24	PC22	PC16	PC18	PC20	PT17		
	2698	PT17	PC16	PP5	PC24	PP4	PT17					
	2687	PT17	PC19	PP4	PC24	PC21	PT17					
	2592	PT17	PC16	PP4	PP5	PC24	PT17					
	2134	PT17	PC17	PC16	PP5	PC22	PC18	PC20	PC23	PT17		
	2077	PT17	PC17	PC19	PC20	PC21	PC23	PC18	PC22	PT17		
	1721	PT17	PC17	PC19	PC18	PC22	PC20	PC21	PC23	PT17		
	1139	PT17	PC23	PC21	PC19	PC17	PT17					
8	3081	PT17	PP5	PP4	PC24	PC22	PC16	PC18	PC20	PC21	PT17	
	2847	PT17	PC17	PC19	PC16	PP4	PC24	PC22	PC18	PC23	PT17	
	2490	PT17	PC17	PC19	PC20	PC21	PC23	PC18	PC22	PC24	PT17	
	2117	PT17	PC17	PC19	PC18	PC22	PC24	PC20	PC21	PC23	PT17	
9	3095	PT17	PP5	PP4	PC24	PC22	PC16	PC18	PC20	PC21	PC23	PT17
	2907	PT17	PC19	PC23	PC20	PC22	PC24	PP5	PP4	PC16	PC17	PT17
	2565	PT17	PC17	PC19	PC20	PC21	PC23	PC18	PC22	PC24	PP5	PT17
	2200	PT17	PC17	PC19	PC18	PC22	PP5	PC24	PC20	PC21	PC23	PT17

Nota. Los puntos se observan en la Figura 37, donde PT_i= Puesto Táctico, PC_i=Punto Crítico, PP_i= Punto Protección.

VI. Discusión

Según lo señala Gutierrez (2018), actualizar constantemente el mapa de delitos de cada comisaria, es acorde con la investigación en curso porque de esa manera se puede plantear mejoras en el patrullaje vehicular para que sea más cercano a las necesidades del día a día. Sin embargo, el autor indica usar solamente la información, para proponer una mejor distribución de vehículos a patrullar puntos críticos seleccionados mediante la aplicación del algoritmo de GRASP, aun así, se presentaría la dificultad de qué rutas deben usar los vehículos para garantizar recorridos óptimos sobre los puntos críticos.

A partir del análisis de resultados, se acepta que independientemente de la cantidad de puntos a visitar en un solo recorrido, cuando se usan algoritmos matemáticos, estos brindan buenas alternativas de rutas para patrullaje, pues diseñan trayectorias en los que la distancia de recorrido sea la menor posible. Estos hallazgos guardan relación con el trabajo de Zevallos (2019) , quien usó otro algoritmo matemático específicamente la programación lineal entera para proponer una ruta de patrullaje para cada sector de los diez que tiene el distrito de San Miguel, cada trayectoria consiste en visitar en un solo recorrido todos los parques y puntos críticos establecidos en el sector correspondiente. Pero en lo que respecta a la técnica utilizada no es conveniente usar la programación lineal porque cada vez que se aumente el número de destinos a visitar, el tiempo que demora en encontrar una solución óptima crece gradualmente, debido a que la búsqueda de la mejor ruta es un problema de optimización de tipo combinatoria, esto implica que para encontrar la mejor solución se debe hacer una exploración extensa de todas las posibles soluciones. Partiendo del argumento expuesto, es mejor el uso de las heurísticas y metaheurísticas puesto que aseguran encontrar rutas para patrullaje vehicular cercanos al óptimo en un tiempo razonable, particularmente en esta investigación hubo un escenario en que el programa diseñó una ruta que involucre visitar dieciséis puntos en un solo recorrido y lo hizo en promedio de cinco a ocho segundos.

En la investigación de Gamba, Domínguez y Navarini (2019) se plantea la idea de construir rutas para patrullaje mediante la aplicación del algoritmo genético dentro de un aplicativo móvil, si bien es cierto el uso de esta metaheurística asegura que en cada intento sus resultados sean distintos por la aleatoriedad dentro de sus procedimientos,

no se debe dejar de lado, que los autores indican que el aplicativo propone automáticamente rutas en que todos los puntos críticos sean visitados en una sola trayectoria, este método podría verse afectado si el número de puntos críticos empieza a aumentar porque las rutas diseñadas serían muy extensas, por ello una mejor alternativa es lo que se hizo en la presenta investigación, en elaborar un aplicativo desktop especializado en el diseño de rutas que le permite al usuario la elección de puntos a visitar, punto de partida y la cantidad máxima de puntos en una ruta, de esta manera los resultados son más aproximados a las decisiones cotidianas.

VII. Conclusiones y Recomendaciones

7.1. Conclusiones

Las rutas o trayectorias empleadas por los vehículos policiales en las rutinas de patrullaje provienen de decisiones empíricas, esto se debe a la dificultad en plantear un plan de rutas, lo que origina que los recorridos sean muy extensos además de que algunas veces las trayectorias sean repetitivas y posiblemente afecten en la frecuencia de patrullaje.

Se planteó un método de construcción de rutas empíricas, debido a que en las comisarías no se realizan registros que precisen la ruta real empleada en las rutinas de patrullaje. El método consistió en presentar cuatro escenarios basados en la cantidad de puntos a visitar.

Se logró construir un aplicativo desktop especializado en diseñar rutas mediante la aplicación de cuatro algoritmos matemático, su uso ofrece una variedad de alternativas según sea la selección de puntos a visitar, el punto de partida y el número máximo de destinos por ruta.

Se hizo una comparación entre las soluciones de los algoritmos matemáticos con las rutas construidas empíricamente. En el análisis se vio que las rutas construidas con los algoritmos indican una distancia de recorrido menor en comparación a las construidas empíricamente.

Se pudo validar las hipótesis específicas de esta investigación, las rutas diseñadas por metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en cada comisaria correspondiente y así mismo se validó la hipótesis general.

Se decidió que las propuestas de rutas sean aquellas que involucren visitar de cinco a nueve destinos, esto porque son decisiones cotidianas en las rutinas de patrullaje. En base a los objetivos específicos los resultados son los siguientes:

- Para la CPNP Piedra Liza se han presentado 40 rutas en el sector S1–1 y 11 rutas en el sector S2-1.
- Para la CPNP Rímac se han presentado 31 rutas en el sector S1, 23 rutas en el sector S2, 48 rutas en el sector S3, 9 rutas en el sector S4 y 11 rutas en el sector S5.
- Para la CPNP Manzano se han presentado 32 rutas en el sector S2, 32 rutas en el sector S3, 36 rutas en el sector S4, 32 rutas en el sector S5 y 36 rutas en el sector S6.
- Para la CPNP Ciudad y Campo se han presentado 23 rutas en el sector S1-2, 24 rutas en el sector S2-1 y 28 rutas en el sector S3-1.
- Para la CPNP Flor de Amancaes se han presentado 36 rutas en el sector S1-2 y 29 rutas en el sector S2-2.

En base al objetivo general diseño 481 rutas para todo el distrito del Rimac.

7.2. Recomendaciones

Para trabajos futuros, considerar la inclusión de elementos como: los horarios específicos donde la actividad delictiva es más recurrente y la priorización de puntos críticos sobre otros. Estos datos pueden integrarse a las técnicas matemáticas con el fin de ofrecer nuevas rutas más cercanas a las necesidades del día a día.

Con respecto al aplicativo desktop hecho para el presente trabajo, las propuestas de rutas mostradas están vinculados a los datos recopilados de enero a junio del presente año, si bien es cierto los resultados cumplen con el objetivo de brindar rutas con recorridos cortos para que los vehículos puedan utilizar para custodiar los puntos críticos y puntos de protección, si se incluyera una conexión con una base datos que permita actualizar los registros de los puntos, los algoritmos matemáticos ofrecerían mayor cantidad de propuestas de rutas.

Se recomienda que los sistemas aplicados a la seguridad ciudadana utilicen herramientas tecnológicas basadas en la aplicación de algoritmos matemáticos para sistematizar las actividades de patrullaje vehicular.

Referencias Bibliográficas

Arias, F. G. (2012). *El Proyecto de Investigación. Introducción a la Metodología Científica. 6ta. Edición*. Fidias G. Arias Odón.

Arteaga Ricci, T. I. (2013). *Aplicación de algoritmos genéticos para el problema de asignación de horarios en la división de ingenierías civil y geomática* [Tesis de Maestria, Universidad Nacional Autónoma de México]. https://repositorio.unam.mx/contenidos/289506

Castañeda Barrios, M. L. (2013). *Aplicación de la metaheurística búsqueda tabú al problema de la ruta más corta para una Empresa Distribuidora de harina de trigo* [Tesis de Bachiller, Universidad Pontificia Bolivariana]. https://repository.upb.edu.co/handle/20.500.11912/587

Chen, H., Cheng, T., & Wise, S. (2015). DESIGNING DAILY PATROL ROUTES FOR POLICING BASED ON ANT COLONY ALGORITHM. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *II-4-W2*, 103-109. https://doi.org/10.5194/isprsannals-II-4-W2-103-2015

Clarke, G., & Wright, J. W. (1964). Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research*, *12*, 568-581. https://doi.org/10.1287/opre.12.4.568

Contreras Pinto, C. M., & Díaz Delgado, M. F. (2010). *METODOS HEURISTICOPARA LA SOLUCION DE PROBLEMAS DE RUTEO DE VEHICULOS CON CAPACIDAD (CVRP)* [Tesis de Pregrado, Universidad Industrial de Santander]. https://uids-primo.hosted.exlibrisgroup.com/permalink/f/1gjmsqs/uids_bucaramanga154380

Cuerpo de Policia Nacional Bolivariana. (2018). *Vigilancia y Patrullaje*. Cuerpo de Policia Nacional Bolivariana. http://www.policianacional.gob.ve/index.php/vigilancia-y-patrullaje/

Fadul, A. O. (2004). *Diseño Estructurado de Algoritmos* (Primera edición). https://es.calameo.com/read/002360365be120298e685

Flores Flores, R. M. (2017). *Mejoramiento del patrullaje integrado PNP – Serenazgo en la jurisdicción de la comisaría PNP Tarapoto con la implementación de una*

solución de inteligencia de negocios [Tesis de Licenciatura, Universidad Nacional de San Martin]. http://repositorio.unsm.edu.pe/handle/11458/2566

Gamba, E. S., Domínguez, V. M., Navarini, A. M., Díaz, D., & Lombardo, V. (2019). Algoritmos Genéticos: Optimización de rutas de patrullaje. *AJEA*, *1*. http://rtyc.utn.edu.ar/index.php/ajea/article/view/528

Garcés Cartagena, Y. F. (Autor), & Zapata Méndez, R. A. (Autor). (2019). *Diseño de un modelo de ruteo de vehículos con consideraciones de inventario en puntos de venta desde un centro de distribución de una empresa del Norte del Valle del Cauca* [Tesis de Titulacion, Universidad del Valle]. https://bibliotecadigital.univalle.edu.co/handle/10893/13805

G.Elías, & Muñoz. (2021). *Delitos contra el Patrimonio y el Orden Socioeconómico*. G.Elías y Muñoz Abogados. https://www.eliasymunozabogados.com/derecho-penal/delitos-contra-patrimonio-orden-socioeconomico

GESTIÓN. (2019a). *Estos son los 10 delitos más denunciados en el Perú en el 2018*. Gestión. https://gestion.pe/fotogalerias/son-10-delitos-denunciados-peru-2018-269447-noticia/

GESTIÓN, N. (2019b). *Estos son los 120 distritos del Perú con mayor delincuencia y violencia del país, según la PNP*. Gestión; NOTICIAS GESTIÓN. https://gestion.pe/peru/policia-detecta-120-distritos-crimenes-violencia-269349-noticia/

Gómez, M. M. (2006). *Introducción a la metodología de la investigación científica*. Editorial Brujas.

Google. (2022). *Acerca de—Google Maps*. https://www.google.com/intl/es/maps/about/#!/

Gutierrez Delgado, M. V. (2018). *Sistema de distribución de carga policial mediante de predicción de delitos* [Tesis de Titulacion, Pontificia Universidad Católica del Perú]. http://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/12939

Hernández, J. L. (1998). *Algoritmos genéticos y su aplicación en optimización de redes* [Tesis de maestria, Universidad Nacional de La Plata]. https://doi.org/10.35537/10915/4111

Hernández Sampieri, R. (2018). *METODOLOGÍA DE LA INVESTIGACIÓN: LAS RUTAS CUANTITATIVA, CUALITATIVA Y MIXTA*. McGraw-Hill Interamericana.

Hillier, F., & Lieberman, G. (2010). *Introduccion a la Investigacion de Operaciones* (novena). McGraw-Hill Interamericana.

Hoyos Gutiérrez, Y. G., & Rubio Gálvez, N. J. (2012). *Diseño e implementación de un algoritmo Metaheurístico Grasp para resolver el problema de ruteo de vehículos con capacidad de carga* [Tesis de Licenciatura, Universidad Nacional de Trujillo]. http://dspace.unitru.edu.pe/handle/UNITRU/11481

Keskin, B. B., Li, S. (Rong), Steil, D., & Spiller, S. (2012). Analysis of an integrated maximum covering and patrol routing problem. *Transportation Research Part E: Logistics and Transportation Review*, *48*(1), 215-232. https://doi.org/10.1016/j.tre.2011.07.005

Labra Díaz, C. (2011). El Modelo de Policía Comunitaria: El caso Chileno. *Revista chilena de DERECHO Y CIENCIA POLITICA*, *2*(1), 49-61.

Lopez, C. A., Mendoza V., J. A., & Cuartas, E. (2008). Algoritmo Para La Exploracion De Todos Los Valores Posibles En El Problema Del Agente Viajero (TPS). *Scientia Et Technica*, *XIV*(39), 399-403.

Martí, R. (2003). *Procedimientos Metaheurísticos en Optimización Combinatoria*. http://www.uv.es/rmarti/paper/docs/heur1.pdf

Microsoft. (2022a). *IDE de Visual Studio 2022: Herramienta de programación para desarrolladores de software*. https://visualstudio.microsoft.com/es/vs/

Microsoft. (2022b). *Software de hojas de cálculo Microsoft Excel | Microsoft 365*. https://www.microsoft.com/es-ww/microsoft-365/excel

MININTER, & Sub Comandancia General PNP. (2018). *GUIA METODOLOGICA PARA EL DISEÑO DE SECTORES Y MAPA DEL DELITO EN LA JURISDICCION DE LAS COMISARIAS DE LA POLICIAL NACIONAL DEL PERU*. [Archivo PDF]. https://cdn.www.gob.pe/uploads/document/file/268370/007-2019-IN__Aprobar_Guia_Metodologica_para_el_dise%C3%B1ode_sectores_y_mapa_del_delito_en_comisarias_de_la_PNP_.pdf

Ministerio de Defensa Nacional Policia Colombia. (2008). *Manual de Patrullaje Urbano*. [Archivo PDF].

http://www2.congreso.gob.pe/sicr/cendocbib/con4_uibd.nsf/430A730DD3D613EB05257AC9005A0432/$FILE/RESOLUCI%C3%93N_COLOMBIA.pdf

Ministerio de Justicia y Derechos Humanos. (2021). *Crímenes Patrimoniales*. https://www.gob.pe/institucion/minjus/informes-publicaciones/1734606-crimenes-patrimoniales

Ministerio Interior. (2018). *Sistema informático de planificación y control del patrullaje SIPCOP se registra en Indecopi*. Plataforma digital única del Estado Peruano. https://www.gob.pe/institucion/mininter/noticias/15813-sistema-informatico-de-planificacion-y-control-del-patrullaje-sipcop-se-registra-en-indecopi

Moreno Bayardo, M. G. (1987). *Introducción a la metodologia de la investigación educativa*. Editorial Progreso.

Nesmachnow, S. (2004). *Algoritmos genéticos paralelos y su aplicación al diseño de redes de comunicaciones confiables* [Tesis de maestría, Universidad de la República (Uruguay). Facultad de Ingeniería. Instituto de Computación – PEDECIBA]. https://hdl.handle.net/20.500.12008/2932

Numbeo. (2021). *Acerca de los Índices de Criminalidad en esta página*. Numbeo. https://es.numbeo.com/criminalidad/%C3%ADndices-explicados

Olivera, A. (2004). Heurísticas para problemas de ruteo de vehículos. *Reportes Técnicos 04-08*. https://hdl.handle.net/20.500.12008/3508

Programa de las Naciones Unidas para el Desarrollo. (2014, abril 15). *Sinopsis: Seguridad Ciudadana*. https://www1.undp.org/content/undp/es/home/librarypage/crisis-prevention-and-recovery/IssueBriefCitizenSecurity.html

Quintero Cuello, D. M., Lahuerta Percipiano, Y., & Moreno Acosta, J. M. (2008). Un índice de criminalidad para Colombia. *Revista Criminalidad*, *50*(1), 37-58.

Rabanal Basalo, P. (2010). *Algoritmos heurísticos y aplicaciones a métodos formales* [Tesis de Doctorado, Universidad Complutense de Madrid]. https://eprints.ucm.es/id/eprint/12027/

Ramírez Rodríguez, L. E. (2016). *Una Solución al Problema de Ruteo de Vehículos Abierto (Ovrp), Implementando la Heurística del Vecino más Cercano.* [Tesis de

Titulacion, Universidad Distrital Francisco Jose Caldas]. http://hdl.handle.net/11349/2985

Real Academia Española. (2020a). *Definición de delito de hurto*. Diccionario panhispánico del español jurídico - Real Academia Española. https://dpej.rae.es/lema/delito-de-hurto

Real Academia Española. (2020b). *Definición de delito de robo*. Diccionario panhispánico del español jurídico - Real Academia Española. https://dpej.rae.es/lema/delito-de-robo

Rentería Ramos, R. R., & Soto Mejía, J. A. (2018). DISEÑO DE UNA SOCIEDAD ARTIFICIAL PARA ESTUDIAR LA MIGRACIÓN FORZADA POR CÓNFLICTO ARMADO INTERNO EN EL SUROCCIDENTE COLOMBIANO. *Investigación Operacional*, *39*(2), 206-219.

Reyes Rueda, J. D. (2020). *Sistema inteligente para la optimización de rutas de vehículos de transporte basado en sistema de información geográfica y metaheurísticas* [Tesis de Máster, Universidad Politécnica de Madrid]. https://oa.upm.es/63691/

Rios Ramirez, R. R. (2017). *Metodología para investigación y redacción* (Primera). Servicios Académicos Intercontinentales S.L. https://www.eumed.net/libros-gratis/2017/1662/index.html

Rodríguez Moguel, E. A. R. (2005). *Metodología de la Investigación*. Univ. J. Autónoma de Tabasco.

Salinero Martín De La Hinojosa, Á. (2016). *DESARROLLO DE ALGORITMOS E INTERFAZ GRÁFICA PARA EL PROBLEMA DE RUTEO DE VEHÍCULOS* [Trabajo de Fin de Grado, Universidad de Sevilla]. https://biblus.us.es/bibing/proyectos/abreproy/91049

Sánchez Hernández, F. (2015). *Aplicación del modelo VRP (Vehicle Routing Problem) para la optimización de una red de distribución* [Tesis de Licenciatura, Universidad Nacional Autónoma de México]. https://repositorio.unam.mx/contenidos/201692

Sanz Torrero, E. (2016). *Técnicas de optimización para la resolución del problema del Job Shop* [Tesis de Grado en Ingeniería en Organización Industrial, Universidad de Valladolid]. http://uvadoc.uva.es/handle/10324/18552

Solomon, M. M. (1987). Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research*, *35*(2), 254-265. https://doi.org/10.1287/opre.35.2.254

Taha, H. (2012). *Investigacion de Operaciones* (novena). Pearson.

Torres Soler, L. C., & Gárzon Torres, N. M. (2016). La inteligencia artificial y la resolución de problemas. *Revista Clepsidra*, *11*(20), 75-88. https://doi.org/10.26564/19001355.628

Vargas Paredes, J., & Penit Granado, V. (2016). *Estudio y aplicación de metaheurísticas y comparación con métodos exhaustivos* [Tesis de Grado, Universidad Complutense Madrid]. https://eprints.ucm.es/39844/

Verizon Connect. (2022). *Que es Optimizacion de Ruta?* Verizon Connect México. https://www.verizonconnect.com/mx/glosario/que-es-optimizacion-de-ruta/

Voss, S. (2001). Meta-heuristics: The State of the Art. *Local Search for Planning and Scheduling*, *2148*, 1-23. https://doi.org/10.1007/3-540-45612-0_1

Yassen, E. T., Arram, A., Ayob, M., Nazri, M., & Nazri, M. (2017). A constructive heuristic for police patrol routing problems. *Pertanika Journal of Science and Technology*, *25*, 87-96.

Yépez, E. (2004). *14 lecciones fundamentales en seguridad ciudadana*. Instituto de Defensa Legal.

Zevallos Murillo, B. A. (2019). *Mejora en la percepción de seguridad ciudadana en los habitantes de un distrito de Lima Metropolitana, mediante la optimización en la asignación de serenos* [Tesis de Licenciatura, Pontificia Universidad Catolica del Perú]. http://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/15234

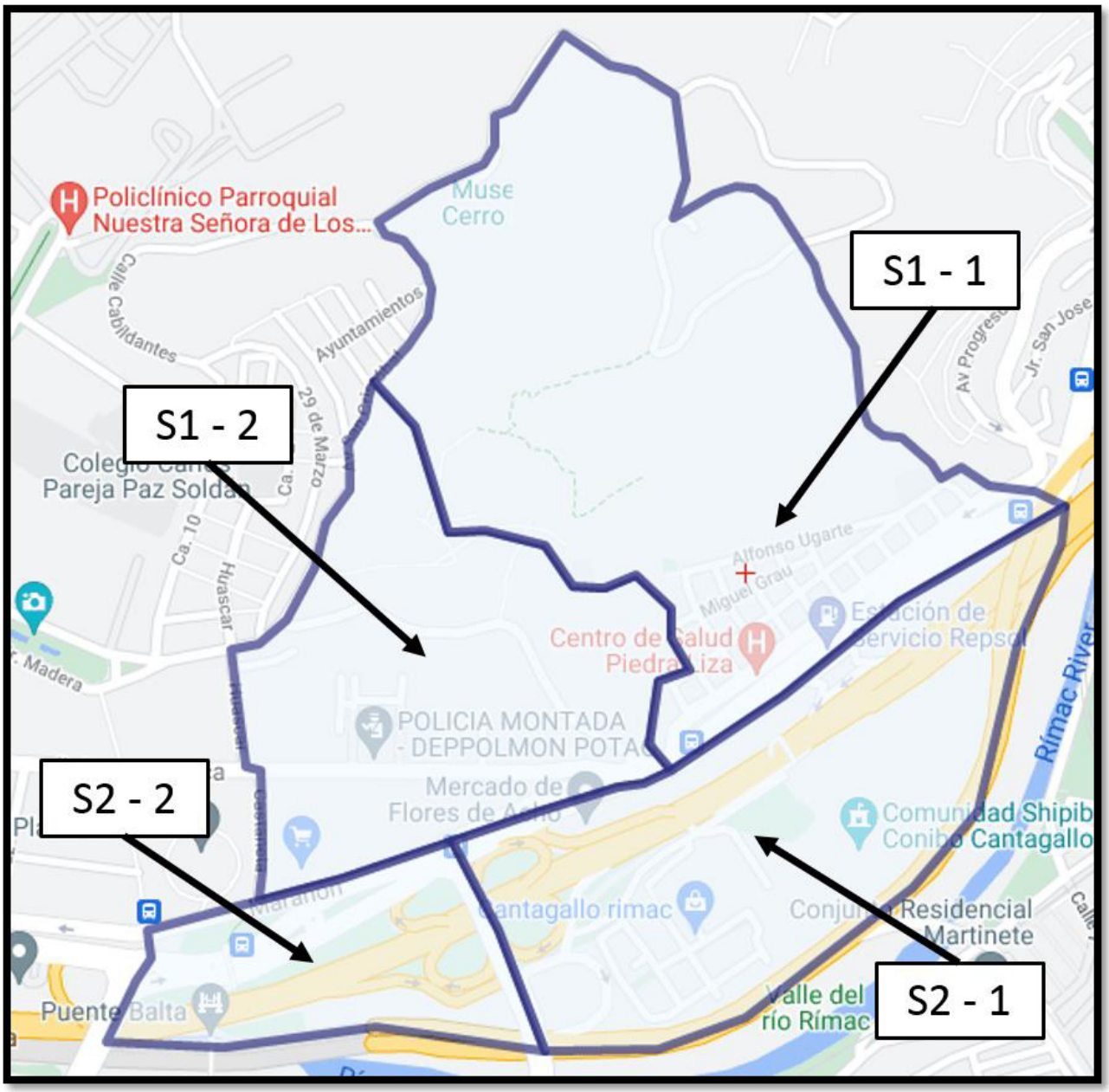
Anexos

Anexo 1. Matriz de Consistencia

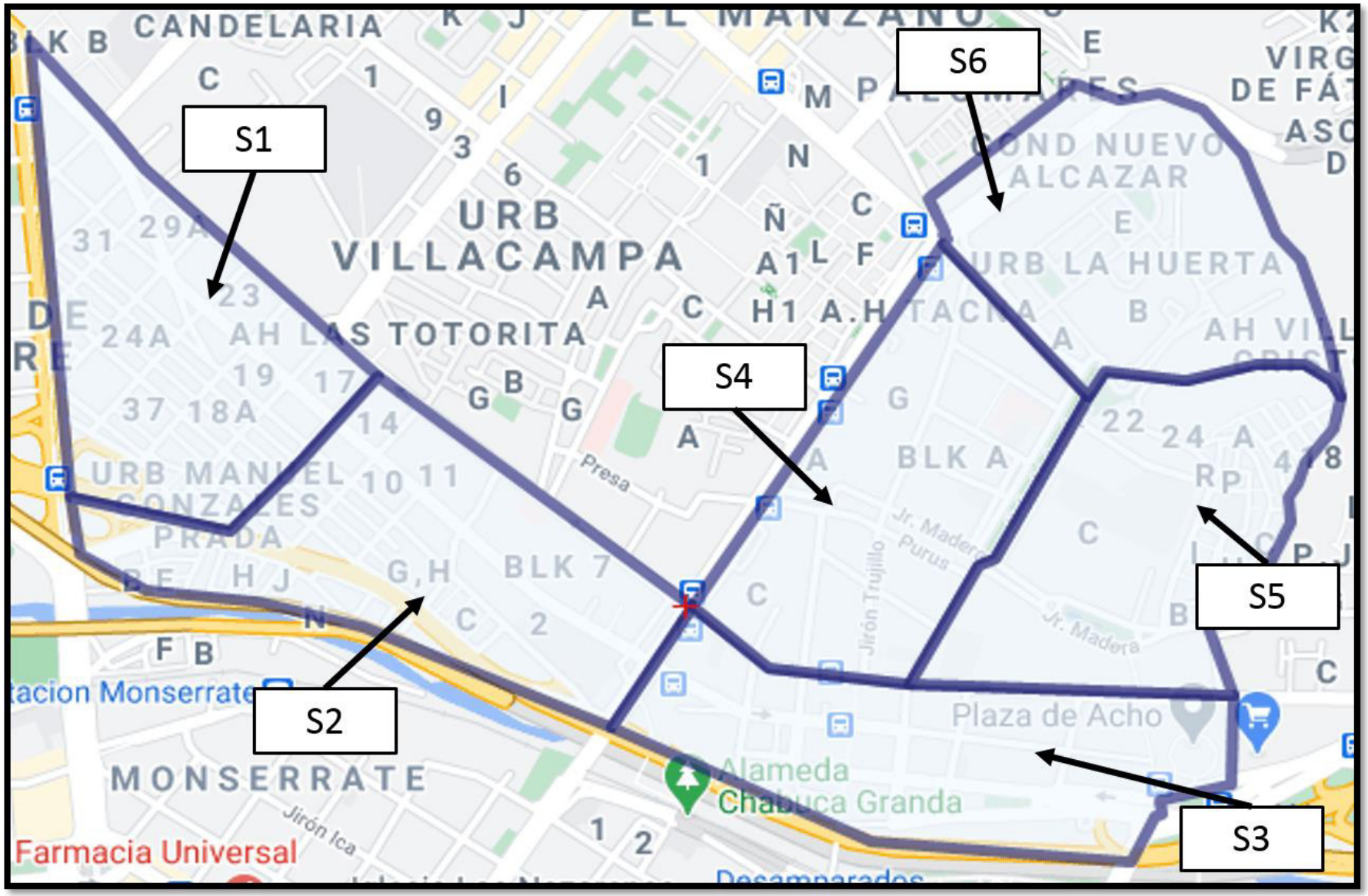
Matriz de Consistencia				
Título: Propuesta de rutas para patrullaje mediante metaheurísticas para optimizar el recorrido de puntos críticos en el distrito del Rímac				
Problema	Objetivo	Hipótesis	Variables, Dimensiones y Indicadores	Metodología
Problema General: ¿Cómo optimizar el recorrido de los vehículos policiales en los puntos críticos de las jurisdicciones de las comisarías del distrito del Rímac con metaheurísticas? Problemas Específicos: 1. ¿Cómo optimizar el recorrido de los vehículos policiales en los puntos críticos de la jurisdicción de la Comisaría PNP Piedra Liza con metaheurísticas? 2. ¿Cómo optimizar el recorrido de los vehículos policiales en los puntos críticos de la jurisdicción de la Comisaría PNP Rímac con metaheurísticas?	Objetivo General: Diseñar rutas para el recorrido óptimo de puntos críticos en las jurisdicciones de las comisarías del distrito de Rímac aplicando metaheurísticas. Objetivos Específicos: 1. Diseñar rutas para el recorrido óptimo de puntos críticos en la jurisdicción de la Comisaria PNP Piedra Liza aplicando metaheurísticas.	Hipótesis General: Las rutas diseñadas por metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en las jurisdicciones de las comisarías del distrito del Rímac. Hipótesis Específicas: 1. Las rutas diseñadas por metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la comisaría PNP Piedra Liza.	Variable Independiente Rutas diseñadas por metaheurísticas Dimensiones: Optimización en la construcción de ruta Indicador: Tiempo Escala de Medición: Numérica (segundos)	Tipo de Investigación Descriptivo y Aplicado Diseño de Investigación No experimental y Transversal Población Información de la actividad del patrullaje en todas las modalidades ya sea en vehiculo ,en moto y a pie, del perido de tiempo de enero a junio del 2021, en cada una de las cinco comisarías del distrito del Rímac.

2. ¿Cómo optimizar el recorrido de los vehículos policiales en los puntos críticos de la jurisdicción de la Comisaría PNP Rímac con metaheurísticas? 3. ¿Cómo optimizar el recorrido de los vehículos policiales en los puntos críticos de la jurisdicción de la Comisaría PNP Manzano con metaheurísticas? 4. ¿Cómo optimizar el recorrido de los vehículos policiales en los puntos críticos de la jurisdicción de la Comisaría PNP Ciudad y Campo con metaheurísticas? 5. ¿Cómo optimizar el recorrido de los vehículos policiales en los puntos críticos de la jurisdicción de la Comisaría PNP Flor de Amancaes con metaheurísticas?	2. Diseñar rutas para el recorrido óptimo de puntos críticos en la jurisdicción de la Comisaria PNP Rímac aplicando metaheurísticas. 3. Diseñar rutas para el recorrido óptimo de puntos críticos en la jurisdicción de la Comisaria PNP Manzano aplicando metaheurísticas. 4. Diseñar rutas para el recorrido óptimo de puntos críticos en la jurisdicción de la Comisaria PNP Ciudad y Campo aplicando metaheurísticas. 5. Diseñar rutas para el recorrido óptimo de puntos críticos en la jurisdicción de la Comisaria PNP Flor de Amancaes aplicando metaheurísticas.	2. Las rutas diseñadas por metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la comisaría PNP Rímac. 3. Las rutas diseñadas por metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la comisaría PNP Manzano. 4. Las rutas diseñadas por metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la comisaría PNP Ciudad y Campo. 5. Las rutas diseñadas por metaheurísticas permiten optimizar el recorrido de los vehículos policiales en los puntos críticos en la jurisdicción de la comisaría PNP Flor de Amancaes.	<u>Variable Dependiente</u> Optimización del recorrido Dimensiones: Optimización en el recorrido de puntos críticos Indicador: Distancia Escala de medición: Numérica (metros)	<u>Muestra</u> Elementos que intervienen en la rutina de patrullaje vehicular: • Los puntos críticos que se identificaron en los mapas de delitos mensuales desde enero hasta junio del año 2021. • Los puntos de protección y puestos tácticos que se identificaron en la hoja de ruta de la primera semana de junio del año 2021. <u>Técnica de Investigación</u> Analisis Documental <u>Instrumento de Investigación</u> Fichas de Trabajo

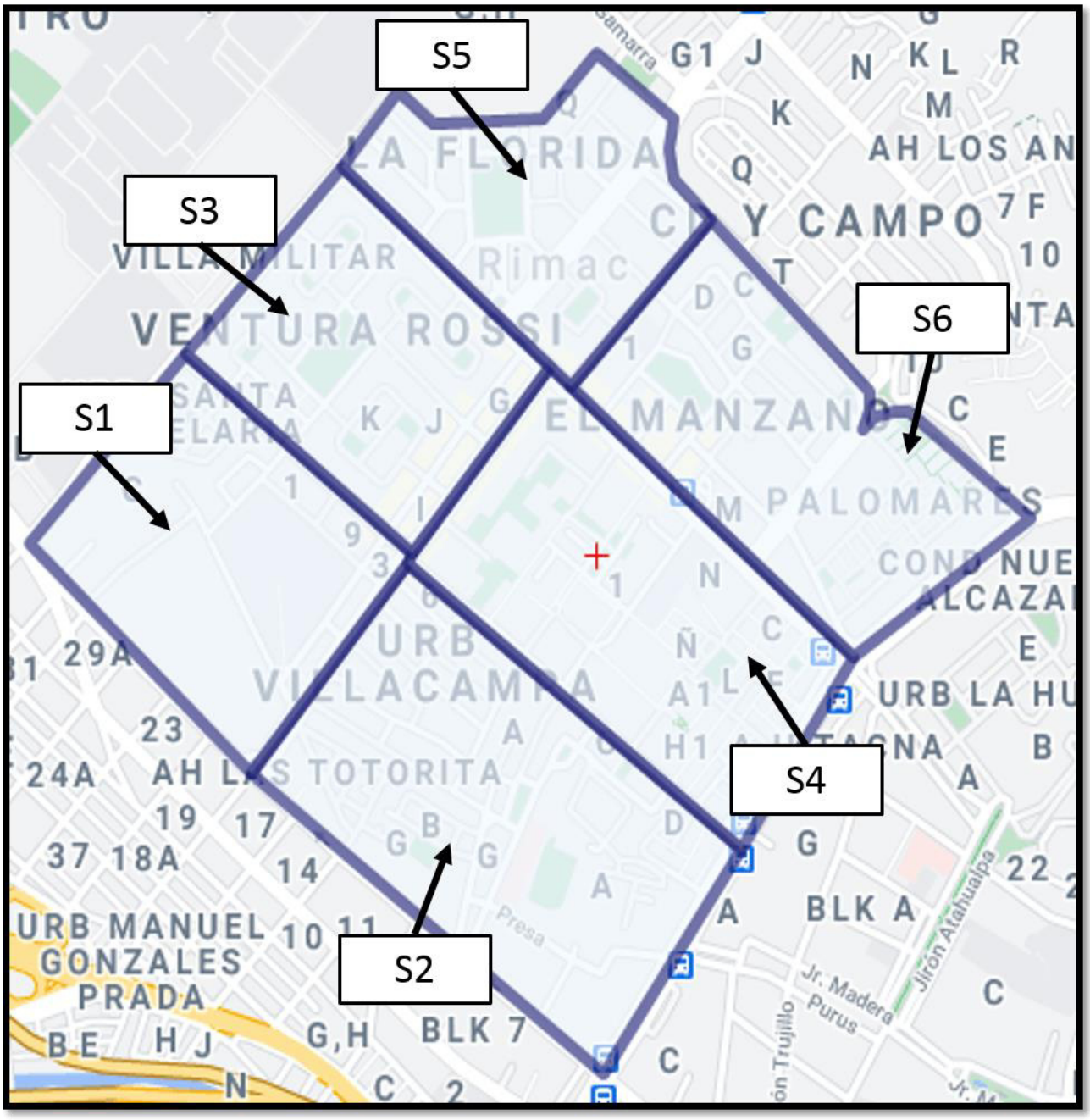
Anexo 2. Mapa de la jurisdicción de CPNP Piedra Liza



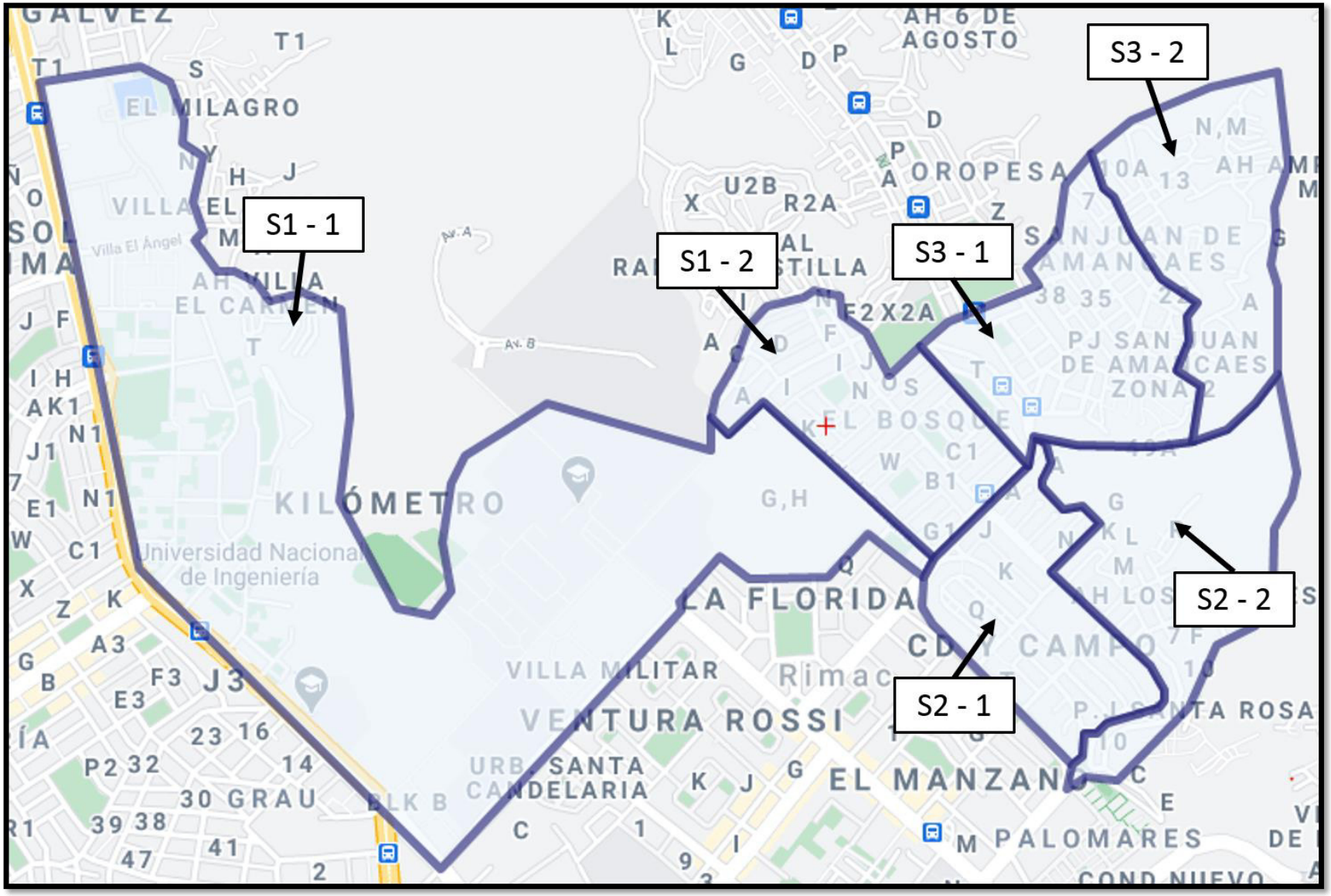
Anexo 3. Mapa de la jurisdicción de CPNP Rimac



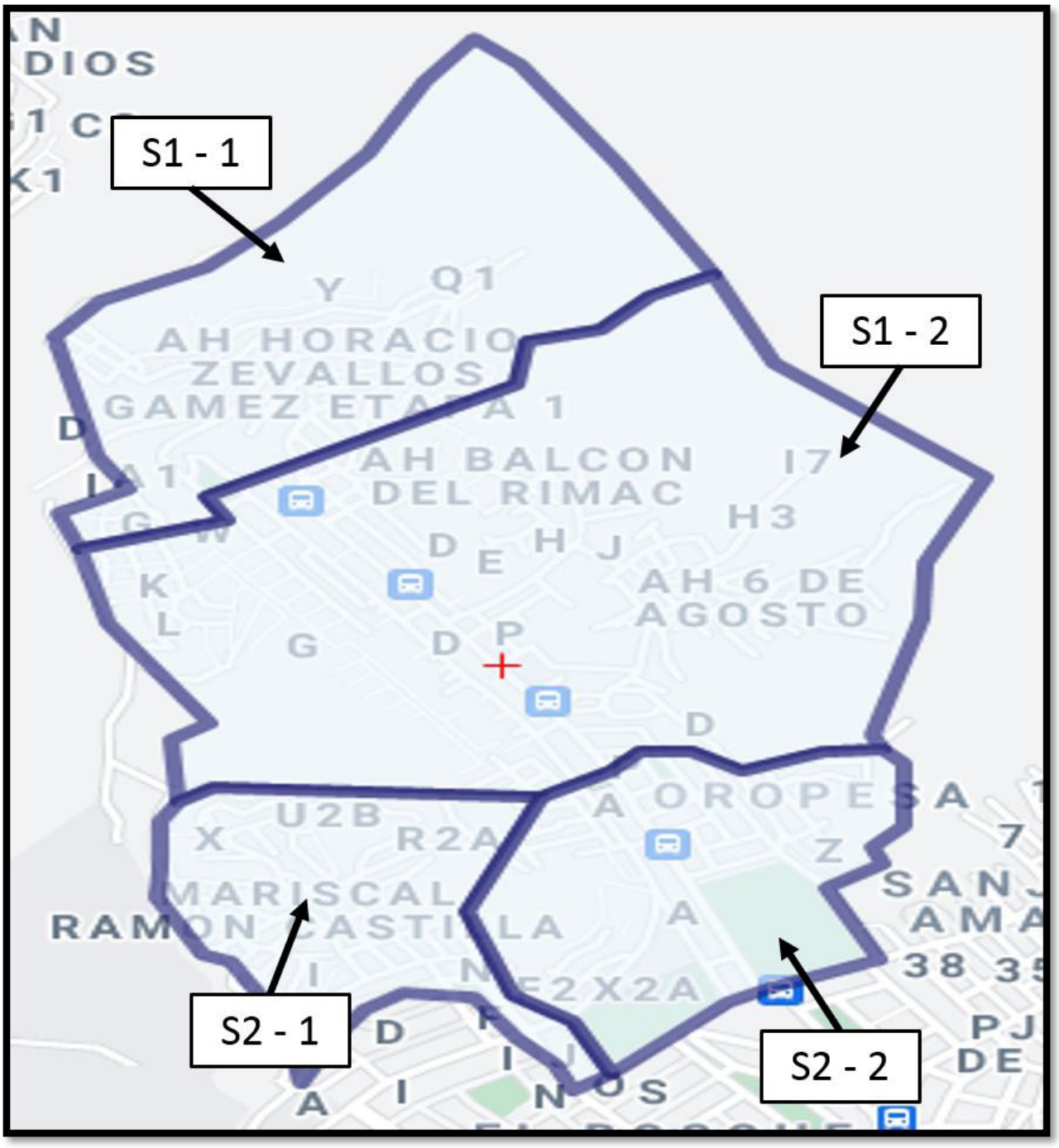
Anexo 4. Mapa de la jurisdicción de CPNP Manzano



Anexo 5. Mapa de la jurisdicción de CPNP Ciudad y Campo



Anexo 6. Mapa de la jurisdicción de CPNP Flor de Amancaes



Anexo 7. Ficha de hoja de ruta

POLICÍA NACIONAL DEL PERÚ

REG.POL.-LIMA **DIVPOL-NORTE-3**

COMISARÍA DE ****** **SEC. OPERACIONES**

HOJA DE RUTA DE PATRULLAJE VEHICULAR DE LA COMISARIA ******
SERVICIO DEL **//** AL **/**/****

VEHÍCULO : **TELÉFONO** **DÍAS**
OPERADOR :
CONDUCTOR:
OPERADOR :
CONDUCTOR:
SECTOR :

HORARIO	PATRULLAJE

ZONA DE PATRULLAJE

PUESTO TÁCTICO

JUNTAS VECINALES

PUNTOS DE PROTECCIÓN

CONTACTO CIUDADANO

INFORMACIÓN PARA EL PATRULLAJE EN EL *** SECTOR - CPNP. *****
CUADERNOS DE CONTROL CIUDADANO
MEDIDAS DE PROTECCIÓN
JUNTAS VECINALES
INSTITUCIONES EDUCATIVAS PÚBLICAS
INSTITUCIONES EDUCATIVAS PRIVADAS
PARROQUIAS
FARMACIAS Y/O BOTICAS

Anexo 8. Instrumentos de recolección de datos

Anexo 8.1. Puntos críticos del área de jurisdicción

Nombre de la Comisaria:			
Sector /Sub sector	Puntos Criticos		Descripción
	N°	Lugar:(Av. Jr. Calle, Psje. Etc)	
	P1		
	P3		
	P4		
	P5		
	P6		
	P7		
	P8		
	P9		
	P10		
	P11		
	P12		
	P13		
	P14		
	P15		
	P16		

Anexo 8.2. Puntos de Protección y Puestos Tácticos

Nombre de la comisaria:					
Sector / Subsector	Puntos Protección		Descripción	Puestos Tacticos	Descripción
	N°	Lugar:(Av. Jr. Calle, Psje. Etc)		Lugar:(Av. Jr. Calle, Psje. Etc)	
	P1				
	P3				
	P4				
	P5				
	P6				
	P7				
	P8				
	P9				
	P10				
	P11				
	P12				
	P13				
	P14				
	P15				
	P16				

Anexo 9. Información Recopilada

Se han suprimido datos en los siguientes formatos para ocultar información confidencial propia de cada comisria.

Anexo 9.1. Comisaria PNP Piedra Liza

Puntos	Coordenadas Geograficas		Descripción
	x	y	
PC1			Local eventos
PC2			Mercado
PC3			Local Comercial
PC4			Local Eventos
PC5			Paradero Bus
PC6			Local Comercial
PC7			Comedor
PC8			Parque
PC9			Mercado
PC10			Local Comercial
PC11			Local Comercial
PC12			Supermercado
PC13			Mercado
PC14			Centro Comercial
PC15			Paradero Bus
PC16			Paradero Bus
PC17			Asociación Viviendas
PC18			Centro Comercial
PC19			Paradero Bus
PC20			Paradero Bus
PC21			Paradero Bus
PP1			Colegio
PP2			Colegio

Puntos	Coordenadas Geograficas		Descripción
	x	y	
PP3			Parroquia
PP4			Botica
PP5			Botica
PP6			Parroquia
PP7			Botica
PP8			Vivienda
PP9			Vivienda
PT1			Puesto Tactico
PT2			Puesto Tactico
PT3			Puesto Tactico
PT4			Puesto Tactico
PT5			Puesto Tactico
PT6			Puesto Tactico
PT7			Puesto Tactico
PT8			Puesto Tactico
PT9			Puesto Tactico
PT10			Puesto Tactico
PT11			Puesto Tactico
PT12			Puesto Tactico
PT13			Puesto Tactico
PT14			Puesto Tactico
PT15			Puesto Tactico

Anexo 9.2. Comisaria PNP Rímac

Puntos	Coordenadas Geograficas		Descripción
	x	y	
PC1			Centro Comercial
PC2			Centro Salud
PC3			Zona Comercial
PC4			Zona Comercial
PC5			Parroquia
PC6			Zona Comercial
PC7			Puente
PC8			Centro Comercial
PC9			Parroquia
PC10			Local Comercial
PC11			Parroquia
PC12			Mercado
PC13			Vecindario
PC14			Paradero Bus
PC15			Centro Comercial
PC16			Vecindario
PC17			Vecindario
PC18			Paradero Bus
PC19			Vecindario
PC20			Zona Comercial
PC21			Zona Comercial
PC22			Vecindario
PC23			Vecindario
PC24			Vecindario
PC25			Vecindario
PC26			Vecindario
PC27			Paradero Bus
PC28			Vecindario
PC29			Parroquia
PC30			Vecindario
PC31			Parque
PC32			Vecindario
PC33			Mercado
PC34			Vecindario
PC35			Parque
PC36			Parque
PC37			Colegio
PC38			Vecindario
PC39			Parroquia
PP1			Zona Comercial
PP2			Centro Comercial
PP3			Mercado
PP4			Colegio

Puntos	Coordenadas Geograficas		Descripción
	x	y	
PP5			Mercado
PP6			Colegio
PP7			Hospital
PP8			Colegio
PP9			Colegio
PP10			Vecindario
PP11			Municipalidad
PP12			Colegio
PP13			Colegio
PP14			Colegio
PP15			Parque
PP16			Parroquia
PP17			Parque
PP18			Parque
PP19			Universidad
PT1			Puesto Tactico
PT2			Puesto Tactico
PT3			Puesto Tactico
PT4			Puesto Tactico
PT5			Puesto Tactico
PT7			Puesto Tactico
PT6			Puesto Tactico
PT8			Puesto Tactico
PT9			Puesto Tactico
PT10			Puesto Tactico
PT11			Puesto Tactico
PT12			Puesto Tactico
PT13			Puesto Tactico
PT14			Puesto Tactico
PT15			Puesto Tactico
PT16			Puesto Tactico
PT17			Puesto Tactico
PT18			Puesto Tactico
PT19			Puesto Tactico
PT20			Puesto Tactico
PT21			Puesto Tactico
PT22			Puesto Tactico
PT23			Puesto Tactico
PT24			Puesto Tactico
PT25			Puesto Tactico

Anexo 9.3. Comisaria PNP Manzano

Puntos	Coordenadas Geograficas		Descripción
	x	y	
PC1			Parque
PC2			Vecindario
PC3			Vecindario
PC4			Vecindario
PC5			Grifo
PC6			Vecindario
PC7			Hospital
PC8			Plaza
PC9			Vecindario
PC10			Vecindario
PC11			Paradero Bus
PC12			Centro Salud
PC13			Zona Comercial
PC14			Vecindario
PC15			Viviendas
PC16			Viviendas
PC17			Parque
PC18			Parque
PC19			Grifo
PC20			Zona Comercial
PC21			Colegio
PC22			Zona Comercial
PC23			Zona Comercial
PC24			Viviendas
PC25			Paradero Bus
PC26			Viviendas
PC27			Viviendas
PC28			Viviendas
PC29			Viviendas
PC30			Viviendas
PC31			Viviendas
PC32			Viviendas
PC33			Viviendas
PC34			Viviendas
PC35			Viviendas
PC36			Viviendas
PC37			Supermercado
PC38			Viviendas
PC39			Zona Comercial
PC40			Colegio
PC41			Parque
PC42			Viviendas
PC43			Viviendas

Puntos	Coordenadas Geograficas		Descripción
	x	y	
PC44			Viviendas
PP1			Centro Salud
PP2			Vecindario
PP3			Parque
PP4			Vecindario
PP5			Mercado
PP6			Banco
PP7			Colegio
PP8			Colegio
PP9			Parroquia
PP10			Centro Salud
PP11			Viviendas
PP12			Parque
PP13			Banco
PP14			Viviendas
PP15			Colegio
PP16			Parroquia
PP17			Banco
PP18			Parque
PT1			Puesto Tactico
PT2			Puesto Tactico
PT3			Puesto Tactico
PT4			Puesto Tactico
PT5			Puesto Tactico
PT6			Puesto Tactico
PT7			Puesto Tactico
PT8			Puesto Tactico
PT9			Puesto Tactico
PT10			Puesto Tactico
PT11			Puesto Tactico
PT12			Puesto Tactico
PT13			Puesto Tactico
PT14			Puesto Tactico
PT15			Puesto Tactico
PT16			Puesto Tactico
PT17			Puesto Tactico
PT18			Puesto Tactico
PT19			Puesto Tactico

Anexo 9.4. Comisaria PNP Ciudad y Campo

Puntos	Coordenadas Geograficas		Descripción
	x	y	
PC1			Zona Comercial
PC2			Zona Comercial
PC3			Plazuela
PC4			Parque
PC5			Parque
PC6			Parque
PC7			Mercado
PC8			Colegio
PC9			Parque
PC10			Mercado
PC11			Parque
PC12			Paradero Bus
PC13			Viviendas
PC14			Mercado
PC15			Zona Comercial
PC16			Viviendas
PC17			Viviendas
PC18			Viviendas
PC19			Viviendas
PP1			Colegio
PP2			Parroquia
PP3			Parque
PP4			Parroquia
PP5			Parroquia
PP6			Parque
PP7			Parque
PP8			Parroquia
PP9			Colegio
PP10			Colegio
PP11			Parroquia
PP12			Viviendas
PP13			Viviendas
PP14			Viviendas
PT1			Puesto Tactico
PT2			Puesto Tactico
PT3			Puesto Tactico
PT4			Puesto Tactico
PT5			Puesto Tactico

Puntos	Coordenadas Geograficas		Descripción
	x	y	
PT6			Puesto Tactico
PT7			Puesto Tactico
PT8			Puesto Tactico
PT9			Puesto Tactico
PT10			Puesto Tactico
PT11			Puesto Tactico
PT12			Puesto Tactico
PT13			Puesto Tactico
PT14			Puesto Tactico
PT15			Puesto Tactico

Anexo 9.5. Comisaria PNP Flor de Amancaes

Puntos	Coordenadas Geograficas		Descripción
	x	y	
PC1			Viviendas
PC2			Viviendas
PC3			Viviendas
PC4			Viviendas
PC5			Viviendas
PC6			Loza Deportiva
PC7			Loza Deportiva
PC8			Viviendas
PC9			Viviendas
PC10			Mercado
PC11			Viviendas
PC12			Viviendas
PC13			Parque
PC14			Parroquia
PC15			Zona Comercial
PC16			Viviendas
PC17			Centro Salud
PC18			Parque
PC19			Colegio
PC20			Viviendas
PC21			Parque
PC22			Parque
PC23			Parque
PC24			Viviendas
PP1			Colegio
PP2			Centro Salud
PP3			Botica
PP4			Parroquia
PP5			Mercado
PT1			Puesto Tactico
PT2			Puesto Tactico
PT3			Puesto Tactico
PT4			Puesto Tactico
PT5			Puesto Tactico
PT6			Puesto Tactico
PT7			Puesto Tactico
PT8			Puesto Tactico
PT9			Puesto Tactico

Puntos	Coordenadas Geograficas		Descripción
	x	y	
PT10			Puesto Tactico
PT11			Puesto Tactico
PT12			Puesto Tactico
PT13			Puesto Tactico
PT14			Puesto Tactico
PT15			Puesto Tactico
PT16			Puesto Tactico
PT17			Puesto Tactico
PT18			Puesto Tactico

Anexo 10. Códigos de programación del aplicativo desktop

Anexo 10.1. Formulario Pantalla_inicio



Los puntos negros indican los controles que están en el formulario, el formulario es llamado *Pantalla_Inicio* ,en (1) está un ComboBox llamado *LisComisarias*, en (2) está un Button llamado *BInicio* y en (3) se ubica un Button llamado *BSalir*. Los códigos de programación son:

```
Imports System.IO
Imports Microsoft.Office.Interop
Imports System.Data
Imports System.Data.OleDb
Imports System
Imports Microsoft.VisualBasic
Public Class Form1

    Private Sub Pantalla_Inicio_Load(sender As Object, e As EventArgs)
Handles MyBase.Load
        LisComisarias.Items.Add("PIEDRA LIZA")
        LisComisarias.Items.Add("RIMAC")
        LisComisarias.Items.Add("MANZANO")
        LisComisarias.Items.Add("CIUDAD Y CAMPO")
        LisComisarias.Items.Add("FLOR DE AMANCAES")
    End Sub

    Private Sub LisComisarias_KeyPress(sender As Object, e As
KeyPressEventArgs) Handles LisComisarias.KeyPress
        e.Handled = True
    End Sub

    Private Sub BSalir_Click(sender As Object, e As EventArgs) Handles
BSalir.Click
        Me.Close()
    End Sub
    Private Sub BInicio_Click(sender As Object, e As EventArgs) Handles
BInicio.Click
        Colocar_Puntos_Coordenadas_Geograficas()
        Select Case LisComisarias.Text
            Case "PIEDRA LIZA"
                Comisaria.NComisaria.Text = "PIEDRA LIZA"
```

```
            Comisaria.Show()
        Case "RIMAC"
            Comisaria.NComisaria.Text = "RIMAC"
            Comisaria.Show()
        Case "MANZANO"
            Comisaria.NComisaria.Text = "MANZANO"
            Comisaria.Show()
        Case "CIUDAD Y CAMPO"
            Comisaria.NComisaria.Text = "CIUDAD Y CAMPO"
            Comisaria.Show()
        Case "FLOR DE AMANCAES"
            Comisaria.NComisaria.Text = "FLOR DE AMANCAES"
            Comisaria.Show()
    End Select
End Sub

Sub Colocar_Puntos_Coordenadas_Geograficas()
    'Puntos criticos de Comisaria de Piedra Liza
    Dim PC_PD As DataTable
    PC_PD = New DataTable()
    PC_PD.Columns.Add("P°")
    PC_PD.Columns.Add("X")
    PC_PD.Columns.Add("Y")
    PC_PD.Columns.Add("Descripcion")
    PC_PD.Rows.Add("PC1", -12.03809, -77.01732, "Local eventos")
    PC_PD.Rows.Add("PC2", -12.04008, -77.01732, "Mercado")
    PC_PD.Rows.Add("PC3", -12.03897, -77.01728, "Local Comercial")
    PC_PD.Rows.Add("PC4", -12.03865, -77.0164, "Local Eventos")
    PC_PD.Rows.Add("PC5", -12.0375, -77.01374, "Paradero de Bus")
    PC_PD.Rows.Add("PC6", -12.03827, -77.01687, "Local Comercial")
    PC_PD.Rows.Add("PC7", -12.03869, -77.01598, "Comedor")
    PC_PD.Rows.Add("PC8", -12.03799, -77.01618, "Parque")
    PC_PD.Rows.Add("PC9", -12.04037, -77.01856, "Mercado")
    PC_PD.Rows.Add("PC10", -12.04096, -77.0199, "Local Comercial")
    PC_PD.Rows.Add("PC11", -12.03873, -77.02228, "Local Comercial")
    PC_PD.Rows.Add("PC12", -12.04039, -77.02252, "Supermercado")
    PC_PD.Rows.Add("PC13", -12.04046, -77.0203, "Mercado")
    PC_PD.Rows.Add("PC14", -12.04104, -77.01758, "Centro Comercial")
    PC_PD.Rows.Add("PC15", -12.0427, -77.01953, "Paradero de Bus")
    PC_PD.Rows.Add("PC16", -12.04337, -77.01902, "Paradero de Bus")
    PC_PD.Rows.Add("PC17", -12.04266, -77.01796, "Asociacion Viviendas")
    PC_PD.Rows.Add("PC18", -12.04234, -77.01746, "Centro Comercial")
    PC_PD.Rows.Add("PC19", -12.04276, -77.02305, "Paradero de Bus")
    PC_PD.Rows.Add("PC20", -12.04302, -77.02297, "Paradero de Bus")
    PC_PD.Rows.Add("PC21", -12.04282, -77.02162, "Paradero de Bus")
    PC_PD.Rows.Add("PP1", -12.03745, -77.01506, "Colegio")
    PC_PD.Rows.Add("PP2", -12.03864, -77.01667, "Colegio")
    PC_PD.Rows.Add("PP3", -12.0375, -77.0152, "Parroquia")
    PC_PD.Rows.Add("PP4", -12.03929, -77.01705, "Botica")
    PC_PD.Rows.Add("PP5", -12.03887, -77.01647, "Botica")
    PC_PD.Rows.Add("PP6", -12.03876, -77.0197, "Parroquia")
    PC_PD.Rows.Add("PP7", -12.03893, -77.02252, "Botica")
    PC_PD.Rows.Add("PP8", -12.04214, -77.01715, "Viviendas")
    PC_PD.Rows.Add("PP9", -12.04279, -77.01728, "Viviendas")
    'Puestos Tacticos
    PC_PD.Rows.Add("PT1", -12.03903, -77.01598, "Puesto Tactico")
    PC_PD.Rows.Add("PT2", -12.03937, -77.01696, "Puesto Tactico")
    PC_PD.Rows.Add("PT3", -12.04031, -77.01763, "Puesto Tactico")
    PC_PD.Rows.Add("PT4", -12.04194, -77.02263, "Puesto Tactico")
    PC_PD.Rows.Add("PT5", -12.03885, -77.02063, "Puesto Tactico")
    PC_PD.Rows.Add("PT6", -12.03659, -77.02148, "Puesto Tactico")
    PC_PD.Rows.Add("PT7", -12.04039, -77.01994, "Puesto Tactico")
    PC_PD.Rows.Add("PT8", -12.04089, -77.01747, "Puesto Tactico")
    PC_PD.Rows.Add("PT9", -12.04311, -77.01938, "Puesto Tactico")
```

```
        PC_PD.Rows.Add("PT10", -12.04321, -77.01794, "Puesto Tactico")
        PC_PD.Rows.Add("PT11", -12.04198, -77.01841, "Puesto Tactico")
        PC_PD.Rows.Add("PT12", -12.04248, -77.01679, "Puesto Tactico")
        PC_PD.Rows.Add("PT13", -12.04254, -77.02376, "Puesto Tactico")
        PC_PD.Rows.Add("PT14", -12.0431, -77.02256, "Puesto Tactico")
        PC_PD.Rows.Add("PT15", -12.04283, -77.01984, "Puesto Tactico")
        'Puntos criticos de Comisaria de Rimac
        Dim PC_RI As DataTable
        PC_RI = New DataTable()
        PC_RI.Columns.Add("P°")
        PC_RI.Columns.Add("X")
        PC_RI.Columns.Add("Y")
        PC_RI.Columns.Add("Descripcion")
        PC_RI.Rows.Add("PC1", -12.0319187024548, -77.0435151, "Centro
Comercial")
        PC_RI.Rows.Add("PC2", -12.0313547024548, -77.0430612, "Centro Salud")
        PC_RI.Rows.Add("PC3", -12.0318362024548, -77.0426249, "Zona
Comercial")
        PC_RI.Rows.Add("PC4", -12.0324533024548, -77.041923, "Zona
Comercial")
        PC_RI.Rows.Add("PC5", -12.0337729024548, -77.0414756, "Parroquia")
        PC_RI.Rows.Add("PC6", -12.0330844024548, -77.0422124, "Zona
Comercial")
        PC_RI.Rows.Add("PC7", -12.0370928024548, -77.0412361, "Puente")
        PC_RI.Rows.Add("PC8", -12.0300209024548, -77.0430279, "Centro
Comercial")
        PC_RI.Rows.Add("PC9", -12.0372697024548, -77.0364265, "Parroquia")
        PC_RI.Rows.Add("PC10", -12.0385781024548, -77.0326727, "Local
Comercial")
        PC_RI.Rows.Add("PC11", -12.0361689024548, -77.0357177, "Parroquia")
        PC_RI.Rows.Add("PC12", -12.0391328024548, -77.0356867, "Mercado")
        PC_RI.Rows.Add("PC13", -12.0381430024548, -77.036639, "Viviendas")
        PC_RI.Rows.Add("PC14", -12.0397334024548, -77.0330497, "Paradero
Bus")
        PC_RI.Rows.Add("PC15", -12.0398952024548, -77.032781, "Centro
Comercial")
        PC_RI.Rows.Add("PC16", -12.0397381024548, -77.0311443, "Viviendas")
        PC_RI.Rows.Add("PC17", -12.0406703024548, -77.0313408, "Viviendas")
        PC_RI.Rows.Add("PC18", -12.0388677024548, -77.0321585, "Paradero
Bus")
        PC_RI.Rows.Add("PC19", -12.0418820024548, -77.0286328, "Viviendas")
        PC_RI.Rows.Add("PC20", -12.0417830024548, -77.029527, "Zona
Comercial")
        PC_RI.Rows.Add("PC21", -12.0408948024548, -77.0293695, "Zona
Comercial")
        PC_RI.Rows.Add("PC22", -12.0399804024548, -77.0302703, "Viviendas")
        PC_RI.Rows.Add("PC23", -12.0414993024548, -77.0304063, "Viviendas")
        PC_RI.Rows.Add("PC24", -12.0422447024548, -77.0238961, "Viviendas")
        PC_RI.Rows.Add("PC25", -12.0404219024548, -77.0241328, "Viviendas")
        PC_RI.Rows.Add("PC26", -12.0415517024548, -77.024021, "Viviendas")
        PC_RI.Rows.Add("PC27", -12.0395611024548, -77.0329212, "Paradero
Bus")
        PC_RI.Rows.Add("PC28", -12.0377395024548, -77.0299232, "Viviendas")
        PC_RI.Rows.Add("PC29", -12.0400175024548, -77.0291709, "Parroquia")
        PC_RI.Rows.Add("PC30", -12.0372834024548, -77.0288929, "Viviendas")
        PC_RI.Rows.Add("PC31", -12.0366951024548, -77.0265854, "Parque")
        PC_RI.Rows.Add("PC32", -12.0369106024548, -77.0287004, "Viviendas")
        PC_RI.Rows.Add("PC33", -12.0366131024548, -77.0307545, "Mercado")
        PC_RI.Rows.Add("PC34", -12.0383764024548, -77.0289809, "Viviendas")
        PC_RI.Rows.Add("PC35", -12.0368318024548, -77.0230222, "Parque")
        PC_RI.Rows.Add("PC36", -12.0379533024548, -77.0267281, "Parque")
        PC_RI.Rows.Add("PC37", -12.0373816024548, -77.0223963, "Colegio")
        PC_RI.Rows.Add("PC38", -12.0360789024548, -77.0220797, "Viviendas")
        PC_RI.Rows.Add("PC39", -12.0364423024548, -77.0216066, "Parroquia")
```

```
        PC_RI.Rows.Add("PP1", -12.0338333024548, -77.0403727, "Zona
Comercial")
        PC_RI.Rows.Add("PP2", -12.0342553024548, -77.0419098, "Centro
Comercial")
        PC_RI.Rows.Add("PP3", -12.0356606024548, -77.0409329, "Mercado")
        PC_RI.Rows.Add("PP4", -12.0331182024548, -77.0409306, "Colegio")
        PC_RI.Rows.Add("PP5", -12.0352691024548, -77.0429595, "Mercado")
        PC_RI.Rows.Add("PP6", -12.0389646024548, -77.0343437, "Colegio")
        PC_RI.Rows.Add("PP7", -12.0367916024548, -77.0348441, "Hospital")
        PC_RI.Rows.Add("PP8", -12.0363061024548, -77.0390889, "Colegio")
        PC_RI.Rows.Add("PP9", -12.0392264024548, -77.0316852, "Colegio")
        PC_RI.Rows.Add("PP10", -12.0410917024548, -77.0274224, "Viviendas")
        PC_RI.Rows.Add("PP11", -12.0420612024548, -77.0268691,
"Municipalidad")
        PC_RI.Rows.Add("PP12", -12.0393028024548, -77.0315788, "Colegio")
        PC_RI.Rows.Add("PP13", -12.0357673024548, -77.0266805, "Colegio")
        PC_RI.Rows.Add("PP14", -12.0365770024548, -77.0268579, "Colegio")
        PC_RI.Rows.Add("PP15", -12.0336853024548, -77.0262649, "Parque")
        PC_RI.Rows.Add("PP16", -12.0352820024548, -77.0253511, "Parroquia")
        PC_RI.Rows.Add("PP17", -12.0387544024548, -77.02563, "Parque")
        PC_RI.Rows.Add("PP18", -12.0344748024548, -77.0249013, "Parque")
        PC_RI.Rows.Add("PP19", -12.0311129024548, -77.0247091, "Universidad")
        PC_RI.Rows.Add("PT1", -12.0325758024548, -77.0416615, "Puesto
Tactico")
        PC_RI.Rows.Add("PT2", -12.0307752024548, -77.0437762, "Puesto
Tactico")
        PC_RI.Rows.Add("PT3", -12.0344770024548, -77.0435711, "Puesto
Tactico")
        PC_RI.Rows.Add("PT4", -12.0352436024548, -77.040306, "Puesto
Tactico")
        PC_RI.Rows.Add("PT5", -12.0326189024548, -77.0432111, "Puesto
Tactico")
        PC_RI.Rows.Add("PT7", -12.0375543024548, -77.0394075, "Puesto
Tactico")
        PC_RI.Rows.Add("PT6", -12.0386001024548, -77.0372315, "Puesto
Tactico")
        PC_RI.Rows.Add("PT8", -12.0379399024548, -77.0396215, "Puesto
Tactico")
        PC_RI.Rows.Add("PT9", -12.0381959024548, -77.0420349, "Puesto
Tactico")
        PC_RI.Rows.Add("PT10", -12.0397092024548, -77.0331308, "Puesto
Tactico")
        PC_RI.Rows.Add("PT11", -12.0395725024548, -77.032701, "Puesto
Tactico")
        PC_RI.Rows.Add("PT12", -12.0420851024548, -77.0232224, "Puesto
Tactico")
        PC_RI.Rows.Add("PT13", -12.0419180024548, -77.0276653, "Puesto
Tactico")
        PC_RI.Rows.Add("PT14", -12.0425312024548, -77.0261422, "Puesto
Tactico")
        PC_RI.Rows.Add("PT15", -12.0425784024548, -77.0276905, "Puesto
Tactico")
        PC_RI.Rows.Add("PT16", -12.0346079024548, -77.0254554, "Puesto
Tactico")
        PC_RI.Rows.Add("PT17", -12.0321398024548, -77.0277735, "Puesto
Tactico")
        PC_RI.Rows.Add("PT18", -12.0387353024548, -77.0321532, "Puesto
Tactico")
        PC_RI.Rows.Add("PT19", -12.0359439024548, -77.0283155, "Puesto
Tactico")
        PC_RI.Rows.Add("PT20", -12.0360795024548, -77.025797, "Puesto
Tactico")
        PC_RI.Rows.Add("PT21", -12.0369222024548, -77.0215351, "Puesto
Tactico")
```

```
        PC_RI.Rows.Add("PT22", -12.0382631024548, -77.0229931, "Puesto
Tactico")
        PC_RI.Rows.Add("PT23", -12.0342277024548, -77.0253348, "Puesto
Tactico")
        PC_RI.Rows.Add("PT24", -12.0320305024548, -77.027557, "Puesto
Tactico")
        PC_RI.Rows.Add("PT25", -12.0315334024548, -77.024208, "Puesto
Tactico")
        'Puntos criticos de Comisaria de Manzano
        Dim PC_MA As DataTable
        PC_MA = New DataTable()
        PC_MA.Columns.Add("P°")
        PC_MA.Columns.Add("X")
        PC_MA.Columns.Add("Y")
        PC_MA.Columns.Add("Descripcion")
        PC_MA.Rows.Add("PC1", -12.0320755024548, -77.0383201, "Parque")
        PC_MA.Rows.Add("PC2", -12.0283787024548, -77.0381684, "Viviendas")
        PC_MA.Rows.Add("PC3", -12.0305030024548, -77.042319, "Viviendas")
        PC_MA.Rows.Add("PC4", -12.0293722024548, -77.0397783, "Viviendas")
        PC_MA.Rows.Add("PC5", -12.0335884024548, -77.0384293, "Grifo")
        PC_MA.Rows.Add("PC6", -12.0329144024548, -77.0379519, "Viviendas")
        PC_MA.Rows.Add("PC7", -12.0345754024548, -77.0340358, "Hospital")
        PC_MA.Rows.Add("PC8", -12.0368108024548, -77.0328598, "Plaza")
        PC_MA.Rows.Add("PC9", -12.0344382024548, -77.0344744, "Viviendas")
        PC_MA.Rows.Add("PC10", -12.0321062024548, -77.0372977, "Viviendas")
        PC_MA.Rows.Add("PC11", -12.0347776024548, -77.029836, "Paradero Bus")
        PC_MA.Rows.Add("PC12", -12.0367918024548, -77.0348439, "Centro
Salud")
        PC_MA.Rows.Add("PC13", -12.0282512024548, -77.0380815, "Zona
Comercial")
        PC_MA.Rows.Add("PC14", -12.0288284024548, -77.035524, "Viviendas")
        PC_MA.Rows.Add("PC15", -12.0247726024548, -77.0359482, "Viviendas")
        PC_MA.Rows.Add("PC16", -12.0256038024548, -77.0388083, "Viviendas")
        PC_MA.Rows.Add("PC17", -12.0269954024548, -77.0380358, "Parque")
        PC_MA.Rows.Add("PC18", -12.0258286024548, -77.0362035, "Parque")
        PC_MA.Rows.Add("PC19", -12.0270130024548, -77.033438, "Grifo")
        PC_MA.Rows.Add("PC20", -12.0278399024548, -77.0324972, "Zona
Comercial")
        PC_MA.Rows.Add("PC21", -12.0299365024548, -77.030964, "Colegio")
        PC_MA.Rows.Add("PC22", -12.0306783024548, -77.0293119, "Zona
Comercial")
        PC_MA.Rows.Add("PC23", -12.0282966024548, -77.0343125, "Zona
Comercial")
        PC_MA.Rows.Add("PC24", -12.0301565024548, -77.0358242, "Viviendas")
        PC_MA.Rows.Add("PC25", -12.0317772024548, -77.0280731, "Paradero
Bus")
        PC_MA.Rows.Add("PC26", -12.0289036024548, -77.0325125, "Viviendas")
        PC_MA.Rows.Add("PC27", -12.0333086024548, -77.029199, "Viviendas")
        PC_MA.Rows.Add("PC28", -12.0342419024548, -77.0305622, "Viviendas")
        PC_MA.Rows.Add("PC29", -12.0236832024548, -77.0315887, "Viviendas")
        PC_MA.Rows.Add("PC30", -12.0245577024548, -77.0326129, "Viviendas")
        PC_MA.Rows.Add("PC31", -12.0228919024548, -77.0365476, "Viviendas")
        PC_MA.Rows.Add("PC32", -12.0228478024548, -77.0311629, "Viviendas")
        PC_MA.Rows.Add("PC33", -12.0259128024548, -77.0344564, "Viviendas")
        PC_MA.Rows.Add("PC34", -12.0228626024548, -77.0337702, "Viviendas")
        PC_MA.Rows.Add("PC35", -12.0249199024548, -77.0347476, "Viviendas")
        PC_MA.Rows.Add("PC36", -12.0234128024548, -77.0348506, "Viviendas")
        PC_MA.Rows.Add("PC37", -12.0287344024548, -77.0312403,
"Supermercado")
        PC_MA.Rows.Add("PC38", -12.0261108024548, -77.030889, "Viviendas")
        PC_MA.Rows.Add("PC39", -12.0272823024548, -77.0328997, "Zona
Comercial")
        PC_MA.Rows.Add("PC40", -12.0303751024548, -77.028444, "Colegio")
        PC_MA.Rows.Add("PC41", -12.0275768024548, -77.0301009, "Parque")
```

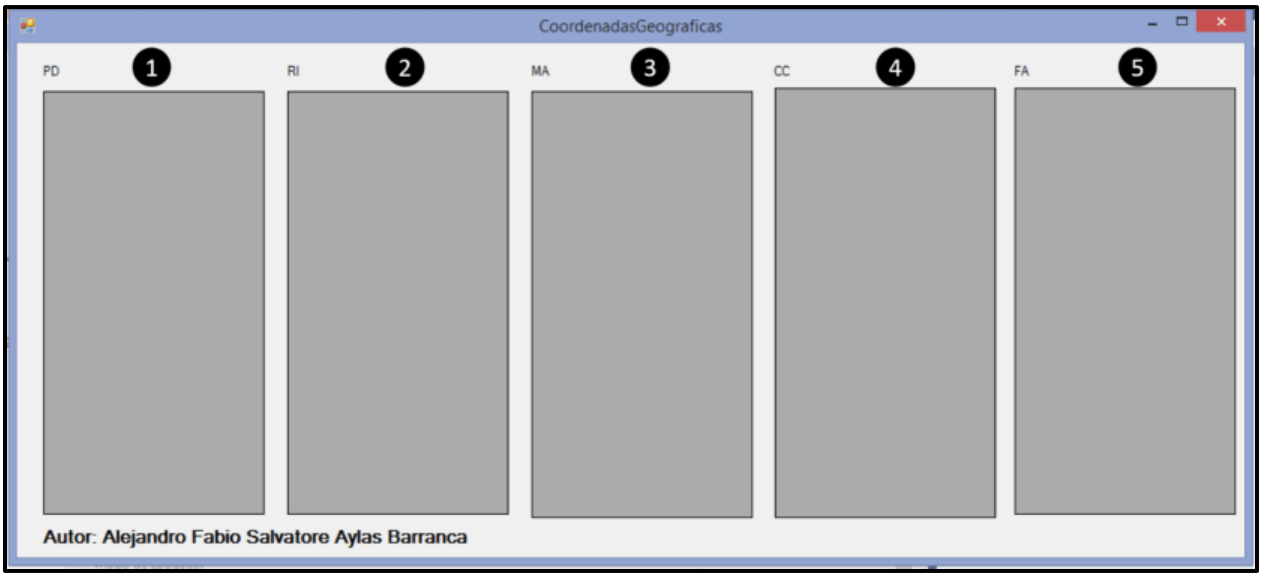
```
        PC_MA.Rows.Add("PC42", -12.0276041024548, -77.0279604, "Viviendas")
        PC_MA.Rows.Add("PC43", -12.0279393024548, -77.027448, "Viviendas")
        PC_MA.Rows.Add("PC44", -12.0245911024548, -77.030427, "Viviendas")
        PC_MA.Rows.Add("PP1", -12.0310076024548, -77.0418398, "Centro Salud")
        PC_MA.Rows.Add("PP2", -12.0329789024548, -77.0345124, "Viviendas")
        PC_MA.Rows.Add("PP3", -12.0332941024548, -77.0327656, "Parque")
        PC_MA.Rows.Add("PP4", -12.0316686024548, -77.0354701, "Viviendas")
        PC_MA.Rows.Add("PP5", -12.0275727024548, -77.0374404, "Mercado")
        PC_MA.Rows.Add("PP6", -12.0295764024548, -77.0355027, "Banco")
        PC_MA.Rows.Add("PP7", -12.0283445024548, -77.036594, "Colegio")
        PC_MA.Rows.Add("PP8", -12.0264842024548, -77.0364236, "Colegio")
        PC_MA.Rows.Add("PP9", -12.0305746024548, -77.0324381, "Parroquia")
        PC_MA.Rows.Add("PP10", -12.0315782024548, -77.0301558, "Centro
Salud")
        PC_MA.Rows.Add("PP11", -12.0316108024548, -77.0325324, "Viviendas")
        PC_MA.Rows.Add("PP12", -12.0263885024548, -77.0327649, "Parque")
        PC_MA.Rows.Add("PP13", -12.0257354024548, -77.0335009, "Banco")
        PC_MA.Rows.Add("PP14", -12.0240747024548, -77.0354928, "Viviendas")
        PC_MA.Rows.Add("PP15", -12.0287164024548, -77.0283731, "Colegio")
        PC_MA.Rows.Add("PP16", -12.0269364024548, -77.031565, "Parroquia")
        PC_MA.Rows.Add("PP17", -12.0276466024548, -77.0324838, "Banco")
        PC_MA.Rows.Add("PP18", -12.0291237024548, -77.0303492, "Parque")
        PC_MA.Rows.Add("PT1", -12.0313548024548, -77.0372308, "Puesto
Tactico")
        PC_MA.Rows.Add("PT2", -12.0279547024548, -77.0410075, "Puesto
Tactico")
        PC_MA.Rows.Add("PT3", -12.0335280024548, -77.034825, "Puesto
Tactico")
        PC_MA.Rows.Add("PT4", -12.0357033024548, -77.0304959, "Puesto
Tactico")
        PC_MA.Rows.Add("PT5", -12.0273526024548, -77.0391857, "Puesto
Tactico")
        PC_MA.Rows.Add("PT6", -12.0259024024548, -77.0370311, "Puesto
Tactico")
        PC_MA.Rows.Add("PT7", -12.0257372024548, -77.0348714, "Puesto
Tactico")
        PC_MA.Rows.Add("PT8", -12.0282611024548, -77.0344278, "Puesto
Tactico")
        PC_MA.Rows.Add("PT9", -12.0321292024548, -77.0306672, "Puesto
Tactico")
        PC_MA.Rows.Add("PT10", -12.0288023024548, -77.0314202, "Puesto
Tactico")
        PC_MA.Rows.Add("PT11", -12.0295113024548, -77.03526, "Puesto
Tactico")
        PC_MA.Rows.Add("PT12", -12.0312113024548, -77.0343989, "Puesto
Tactico")
        PC_MA.Rows.Add("PT13", -12.0234015024548, -77.0329634, "Puesto
Tactico")
        PC_MA.Rows.Add("PT14", -12.0252475024548, -77.033245, "Puesto
Tactico")
        PC_MA.Rows.Add("PT15", -12.0234361024548, -77.0354881, "Puesto
Tactico")
        PC_MA.Rows.Add("PT16", -12.0248090024548, -77.0356748, "Puesto
Tactico")
        PC_MA.Rows.Add("PT17", -12.0296746024548, -77.0290047, "Puesto
Tactico")
        PC_MA.Rows.Add("PT18", -12.0265276024548, -77.0292146, "Puesto
Tactico")
        PC_MA.Rows.Add("PT19", -12.0278076024548, -77.0323197, "Puesto
Tactico")
        'Puntos criticos de Comisaria de Ciudad y Campo
        Dim PC_CC As DataTable
        PC_CC = New DataTable()
        PC_CC.Columns.Add("P°")
```

```
        PC_CC.Columns.Add("X")
        PC_CC.Columns.Add("Y")
        PC_CC.Columns.Add("Descripcion")
        PC_CC.Rows.Add("PC1", -12.0109221024548, -77.0527445, "Zona
Comercial")
        PC_CC.Rows.Add("PC2", -12.0207325024548, -77.0300121, "Zona
Comercial")
        PC_CC.Rows.Add("PC3", -12.0163364024548, -77.0339128, "Plazuela")
        PC_CC.Rows.Add("PC4", -12.0188031024548, -77.0332468, "Parque")
        PC_CC.Rows.Add("PC5", -12.0189637024548, -77.0309556, "Parque")
        PC_CC.Rows.Add("PC6", -12.0184739024548, -77.0294347, "Parque")
        PC_CC.Rows.Add("PC7", -12.0156244024548, -77.0266535, "Mercado")
        PC_CC.Rows.Add("PC8", -12.0162187024548, -77.0270489, "Colegio")
        PC_CC.Rows.Add("PC9", -12.0166504024548, -77.0291132, "Parque")
        PC_CC.Rows.Add("PC10", -12.0172978024548, -77.0272777, "Mercado")
        PC_CC.Rows.Add("PC11", -12.0155176024548, -77.0234203, "Parque")
        PC_CC.Rows.Add("PC12", -12.0143012024548, -77.0250945, "Paradero
Bus")
        PC_CC.Rows.Add("PC13", -12.0218011024548, -77.0300849, "Viviendas")
        PC_CC.Rows.Add("PC14", -12.0255593024548, -77.0280853, "Mercado")
        PC_CC.Rows.Add("PC15", -12.0240268024548, -77.0266329, "Zona
Comercial ")
        PC_CC.Rows.Add("PC16", -12.0215378024548, -77.0291719, "Viviendas")
        PC_CC.Rows.Add("PC17", -12.0245685024548, -77.0292837, "Viviendas")
        PC_CC.Rows.Add("PC18", -12.0274993024548, -77.0266092, "Viviendas")
        PC_CC.Rows.Add("PC19", -12.0231260024548, -77.0255939, "Viviendas")
        PC_CC.Rows.Add("PP1", -12.0189806024548, -77.0354291, "Colegio")
        PC_CC.Rows.Add("PP2", -12.0222677024548, -77.0307396, "Parroquia")
        PC_CC.Rows.Add("PP3", -12.0179936024548, -77.030545, "Parque")
        PC_CC.Rows.Add("PP4", -12.0213093024548, -77.0318748, "Parroquia")
        PC_CC.Rows.Add("PP5", -12.0178373024548, -77.0340964, "Parroquia")
        PC_CC.Rows.Add("PP6", -12.0182225024548, -77.0286056, "Parque")
        PC_CC.Rows.Add("PP7", -12.0184851024548, -77.0270627, "Parque")
        PC_CC.Rows.Add("PP8", -12.0159272024548, -77.0284724, "Parroquia")
        PC_CC.Rows.Add("PP9", -12.0157281024548, -77.0257038, "Colegio")
        PC_CC.Rows.Add("PP10", -12.0234870024548, -77.0271348, "Colegio")
        PC_CC.Rows.Add("PP11", -12.0236123024548, -77.0283606, "Parroquia")
        PC_CC.Rows.Add("PP12", -12.0229374024548, -77.0303892, "Viviendas")
        PC_CC.Rows.Add("PP13", -12.0217138024548, -77.0269619, "Viviendas")
        PC_CC.Rows.Add("PP14", -12.0259806024548, -77.025301, "Viviendas")
        PC_CC.Rows.Add("PT1", -12.0179739024548, -77.05086, "Puesto Tactico")
        PC_CC.Rows.Add("PT2", -12.0263997024548, -77.0462222, "Puesto
Tactico")
        PC_CC.Rows.Add("PT3", -12.0200993024548, -77.0305818, "Puesto
Tactico")
        PC_CC.Rows.Add("PT4", -12.0170339024548, -77.0324617, "Puesto
Tactico")
        PC_CC.Rows.Add("PT5", -12.0199492024548, -77.0333958, "Puesto
Tactico")
        PC_CC.Rows.Add("PT6", -12.0192880024548, -77.0289841, "Puesto
Tactico")
        PC_CC.Rows.Add("PT7", -12.0148431024548, -77.0264114, "Puesto
Tactico")
        PC_CC.Rows.Add("PT8", -12.0168033024548, -77.0266012, "Puesto
Tactico")
        PC_CC.Rows.Add("PT9", -12.0135925024548, -77.0255737, "Puesto
Tactico")
        PC_CC.Rows.Add("PT10", -12.0146493024548, -77.0242101, "Puesto
Tactico")
        PC_CC.Rows.Add("PT11", -12.0143659024548, -77.0225512, "Puesto
Tactico")
        PC_CC.Rows.Add("PT12", -12.0266250024548, -77.0270077, "Puesto
Tactico")
```

```
        PC_CC.Rows.Add("PT13", -12.0224211024548, -77.0303281, "Puesto
Tactico")
        PC_CC.Rows.Add("PT14", -12.0240315024548, -77.0251618, "Puesto
Tactico")
        PC_CC.Rows.Add("PT15", -12.0276795024548, -77.0267756, "Puesto
Tactico")
        'Puntos criticos de Comisaria de Flor de Amancaes
        Dim PC_FA As DataTable
        PC_FA = New DataTable()
        PC_FA.Columns.Add("P°")
        PC_FA.Columns.Add("X")
        PC_FA.Columns.Add("Y")
        PC_FA.Columns.Add("Descripcion")
        PC_FA.Rows.Add("PC1", -12.0071350024548, -77.0373339, "Viviendas")
        PC_FA.Rows.Add("PC2", -12.0063669024548, -77.0366321, "Viviendas")
        PC_FA.Rows.Add("PC3", -12.0054558024548, -77.0342717, "Viviendas")
        PC_FA.Rows.Add("PC4", -12.0123184024548, -77.0320679, "Viviendas")
        PC_FA.Rows.Add("PC5", -12.0120267024548, -77.0327753, "Viviendas")
        PC_FA.Rows.Add("PC6", -12.0079212024548, -77.0364155, "Loza
Deportiva")
        PC_FA.Rows.Add("PC7", -12.0113205024548, -77.0319483, "Loza
Deportiva")
        PC_FA.Rows.Add("PC8", -12.0078860024548, -77.0346172, "Viviendas")
        PC_FA.Rows.Add("PC9", -12.0091653024548, -77.0341663, "Viviendas")
        PC_FA.Rows.Add("PC10", -12.0086488024548, -77.0355521, "Mercado")
        PC_FA.Rows.Add("PC11", -12.0099127024548, -77.0343001, "Viviendas")
        PC_FA.Rows.Add("PC12", -12.0092670024548, -77.0327929, "Viviendas")
        PC_FA.Rows.Add("PC13", -12.0163341024548, -77.0339078, "Parque")
        PC_FA.Rows.Add("PC14", -12.0144777024548, -77.0357882, "Parroquia")
        PC_FA.Rows.Add("PC15", -12.0159293024548, -77.0338668, "Zona
Comercial")
        PC_FA.Rows.Add("PC16", -12.0166533024548, -77.03043, "Viviendas")
        PC_FA.Rows.Add("PC17", -12.0178908024548, -77.0318421, "Centro
Salud")
        PC_FA.Rows.Add("PC18", -12.0145571024548, -77.0309482, "Parque")
        PC_FA.Rows.Add("PC19", -12.0159382024548, -77.0316092, "Colegio")
        PC_FA.Rows.Add("PC20", -12.0133527024548, -77.0312522, "Viviendas")
        PC_FA.Rows.Add("PC21", -12.0123543024548, -77.0315349, "Parque")
        PC_FA.Rows.Add("PC22", -12.0139136024548, -77.0304233, "Parque")
        PC_FA.Rows.Add("PC23", -12.0128492024548, -77.031792, "Parque")
        PC_FA.Rows.Add("PC24", -12.0129142024548, -77.0298745, "Viviendas")
        PC_FA.Rows.Add("PP1", -12.0110907024548, -77.0330118, "Colegio")
        PC_FA.Rows.Add("PP2", -12.0080691024548, -77.0354148, "Centro Salud")
        PC_FA.Rows.Add("PP3", -12.0106096024548, -77.0339447, "Botica")
        PC_FA.Rows.Add("PP4", -12.0154158024548, -77.0288776, "Parroquia")
        PC_FA.Rows.Add("PP5", -12.0138618024548, -77.0298129, "Mercado")
        PC_FA.Rows.Add("PT1", -12.0060585024548, -77.0348021, "Puesto
Tactico")
        PC_FA.Rows.Add("PT2", -12.0054182024548, -77.0376804, "Puesto
Tactico")
        PC_FA.Rows.Add("PT3", -12.0044230024548, -77.0338304, "Puesto
Tactico")
        PC_FA.Rows.Add("PT4", -12.0075364024548, -77.0364415, "Puesto
Tactico")
        PC_FA.Rows.Add("PT5", -12.0082850024548, -77.0353281, "Puesto
Tactico")
        PC_FA.Rows.Add("PT6", -12.0107968024548, -77.033245, "Puesto
Tactico")
        PC_FA.Rows.Add("PT7", -12.0115252024548, -77.0312519, "Puesto
Tactico")
        PC_FA.Rows.Add("PT8", -12.0108890024548, -77.0315407, "Puesto
Tactico")
        PC_FA.Rows.Add("PT9", -12.0075203024548, -77.034993, "Puesto
Tactico")
```

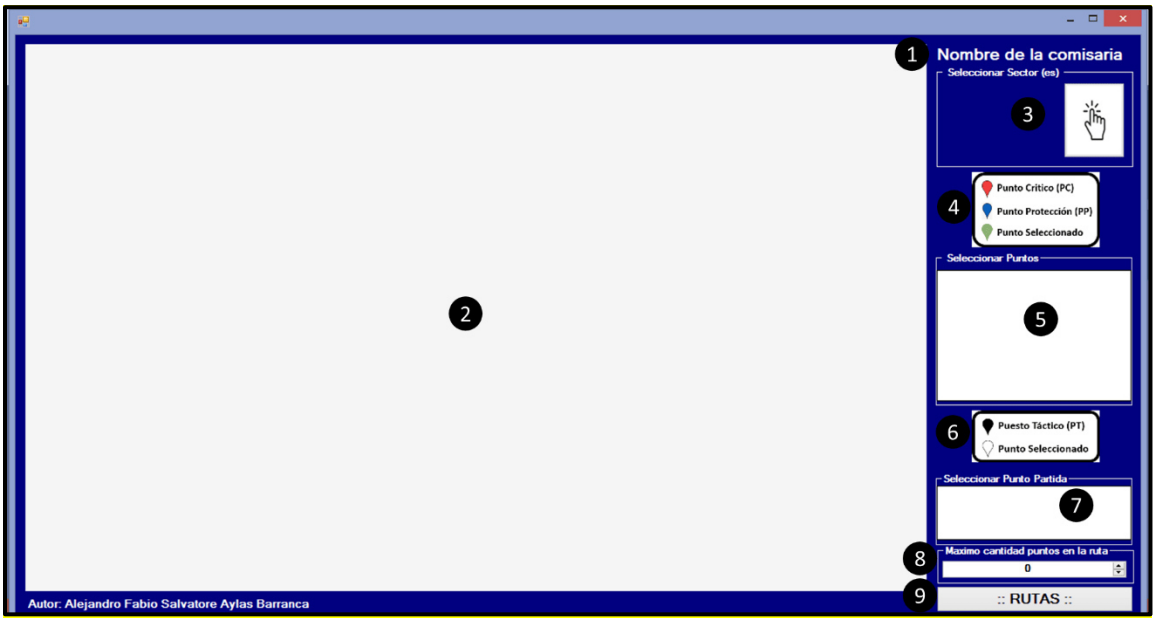
```
        PC_FA.Rows.Add("PT10", -12.0163097024548, -77.0348469, "Puesto
Tactico")
        PC_FA.Rows.Add("PT11", -12.0172053024548, -77.0328874, "Puesto
Tactico")
        PC_FA.Rows.Add("PT12", -12.0155559024548, -77.0344915, "Puesto
Tactico")
        PC_FA.Rows.Add("PT13", -12.0148849024548, -77.0354433, "Puesto
Tactico")
        PC_FA.Rows.Add("PT14", -12.0159192024548, -77.030356, "Puesto
Tactico")
        PC_FA.Rows.Add("PT15", -12.0140889024548, -77.0298482, "Puesto
Tactico")
        PC_FA.Rows.Add("PT16", -12.0148111024548, -77.0312934, "Puesto
Tactico")
        PC_FA.Rows.Add("PT17", -12.0169729024548, -77.0323834, "Puesto
Tactico")
        PC_FA.Rows.Add("PT18", -12.0129046024548, -77.031247, "Puesto
Tactico")
        CoordenadasGeograficas.DataGridView1.DataSource = PC_PD
        CoordenadasGeograficas.DataGridView2.DataSource = PC_RI
        CoordenadasGeograficas.DataGridView3.DataSource = PC_MA
        CoordenadasGeograficas.DataGridView4.DataSource = PC_CC
        CoordenadasGeograficas.DataGridView5.DataSource = PC_FA
    End Sub
End Class
```

Anexo 10.2. Formulario CoordenadasGeograficas



Los controles asociados al formulario son: en (1) se ubica en el *DataGridView1* lo referido a CPNP Piedra Liza, en (2) se ubica en el *DataGridView2* lo referido a CPNP Rímac, en (3) se ubica en el *DataGridView3* lo referido a CPNP Manzano, en (4) se ubica en el *DataGridView4* lo referido a CPNP Ciudad y Campo, en (5) se ubica en el *DataGridView5* lo referido a CPNP Flor de Amancaes.

Anexo 10.3. Formulario Comisaria



Este formulario es llamado *Comisaria* los controles presentes son: en (1) está un Label llamado *NComisaria*, en (2) se ubica un GMapControl llamado *Mapa,* en (3) se encuentra un GroupBox llamado *GroupBoxSelecSector*, este contiene un CheckedListBox llamado *CheckedListBoxSectores* y un Button llamado *BSector,* en (4) se encuentra *PictureBox1,* en (5) un GroupBox llamado *GroupBoxSelecPuntos* que contiene un DataGridView llamado *PCriticos,* en (6) se encuentra *PictureBox2,* en (7) un GroupBox llamado *GroupBoxIndicacion* que contiene un DataGridView llamado *PTactico, en (8)* está un GroupBox llamado *GroupBox1* que contiene un NumericUpDown llamado *NumPuntos* y en (9) se encuentra un Button llamado *BEjecutar*. Los códigos de programación son:

```
Imports GMap.NET
Imports GMap.NET.MapProviders
Imports GMap.NET.WindowsForms
Imports GMap.NET.WindowsForms.Markers

Public Class Comisaria
    Dim marker As GMarkerGoogle
    Dim markerOverlay As GMapOverlay
    Private Sub Comisaria_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        Me.Location = Screen.PrimaryScreen.WorkingArea.Location
        Me.Size = Screen.PrimaryScreen.WorkingArea.Size
        Select Case NComisaria.Text
            Case "PIEDRA LIZA"
                Poligono_PD()
                CheckedListBoxSectores.Items.Clear()
                CheckedListBoxSectores.Items.Add("S1-1")
                CheckedListBoxSectores.Items.Add("S1-2")
```

```
                CheckedListBoxSectores.Items.Add("S2-1")
                CheckedListBoxSectores.Items.Add("S2-2")
            Case "RIMAC"
                Poligono_RI()
                CheckedListBoxSectores.Items.Clear()
                CheckedListBoxSectores.Items.Add("S1")
                CheckedListBoxSectores.Items.Add("S2")
                CheckedListBoxSectores.Items.Add("S3")
                CheckedListBoxSectores.Items.Add("S4")
                CheckedListBoxSectores.Items.Add("S5")
                CheckedListBoxSectores.Items.Add("S6")
            Case "MANZANO"
                Poligono_MA()
                CheckedListBoxSectores.Items.Clear()
                CheckedListBoxSectores.Items.Add("S1")
                CheckedListBoxSectores.Items.Add("S2")
                CheckedListBoxSectores.Items.Add("S3")
                CheckedListBoxSectores.Items.Add("S4")
                CheckedListBoxSectores.Items.Add("S5")
                CheckedListBoxSectores.Items.Add("S6")
            Case "CIUDAD Y CAMPO"
                Poligono_CC()
                CheckedListBoxSectores.Items.Clear()
                CheckedListBoxSectores.Items.Add("S1-1")
                CheckedListBoxSectores.Items.Add("S1-2")
                CheckedListBoxSectores.Items.Add("S2-1")
                CheckedListBoxSectores.Items.Add("S2-2")
                CheckedListBoxSectores.Items.Add("S3-1")
                CheckedListBoxSectores.Items.Add("S3-2")
            Case "FLOR DE AMANCAES"
                Poligono_FA()
                CheckedListBoxSectores.Items.Clear()
                CheckedListBoxSectores.Items.Add("S1-1")
                CheckedListBoxSectores.Items.Add("S1-2")
                CheckedListBoxSectores.Items.Add("S2-1")
                CheckedListBoxSectores.Items.Add("S2-2")
        End Select
    End Sub
    Sub Poligono_MA()
        Mapa.DragButton = MouseButtons.Left
        Mapa.CanDragMap = True
        Mapa.MapProvider = GMapProviders.GoogleMap
        Mapa.Position = New PointLatLng(-12.03009, -77.03245)
        Mapa.MinZoom = 0
        Mapa.MaxZoom = 24
        Mapa.Zoom = 15
        Mapa.AutoScroll = True
        'Agregando los poligonos del area de jurisdiccion
        Dim S1 As DataTable
        S1 = New DataTable()
        S1.Columns.Add("")
        S1.Columns.Add("")
        S1.Rows.Add(-12.03867, -77.03232)
        S1.Rows.Add(-12.03371, -77.03879)
        S1.Rows.Add(-12.0302, -77.0359)
        S1.Rows.Add(-12.03494, -77.02979)
        S1.Rows.Add(-12.03867, -77.03232)
        Dim Poligono_S1 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        Dim lat As Double
        Dim lng As Double
        For i = 0 To S1.Rows.Count - 1
```

```
            lat = Convert.ToDouble(S1(i)(0))
            lng = Convert.ToDouble(S1(i)(1))
            puntos.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto As GMapPolygon = New GMapPolygon(puntos,
"Poligono")
        Poligono_S1.Polygons.Add(poligonosPunto)
        Mapa.Overlays.Add(Poligono_S1)
        Dim S2 As DataTable
        S2 = New DataTable()
        S2.Columns.Add("")
        S2.Columns.Add("")
        S2.Rows.Add(-12.03371, -77.03879)
        S2.Rows.Add(-12.02988, -77.04287)
        S2.Rows.Add(-12.02673, -77.04)
        S2.Rows.Add(-12.03005, -77.03582)
        S2.Rows.Add(-12.0302, -77.0359)
        S2.Rows.Add(-12.03371, -77.03879)
        Dim Poligono_S2 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos2 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S2.Rows.Count - 1
            lat = Convert.ToDouble(S2(i)(0))
            lng = Convert.ToDouble(S2(i)(1))
            puntos2.Add(New PointLatLng(lat, lng))

        Next
        Dim poligonosPunto2 As GMapPolygon = New GMapPolygon(puntos2,
"Poligono")
        Poligono_S2.Polygons.Add(poligonosPunto2)
        Mapa.Overlays.Add(Poligono_S2)
        Dim S3 As DataTable
        S3 = New DataTable()
        S3.Columns.Add("")
        S3.Columns.Add("")
        S3.Rows.Add(-12.03494, -77.02979)
        S3.Rows.Add(-12.0302, -77.0359)
        S3.Rows.Add(-12.03005, -77.03582)
        S3.Rows.Add(-12.02695, -77.03331)
        S3.Rows.Add(-12.02729, -77.03292)
        S3.Rows.Add(-12.03178, -77.02771)
        S3.Rows.Add(-12.03494, -77.02979)
        Dim Poligono_S3 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos3 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S3.Rows.Count - 1
            lat = Convert.ToDouble(S3(i)(0))
            lng = Convert.ToDouble(S3(i)(1))
            puntos3.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto3 As GMapPolygon = New GMapPolygon(puntos3,
"Poligono")
        Poligono_S3.Polygons.Add(poligonosPunto3)
        Mapa.Overlays.Add(Poligono_S3)
        Dim S4 As DataTable
        S4 = New DataTable()
        S4.Columns.Add("")
        S4.Columns.Add("")
        S4.Rows.Add(-12.03005, -77.03582)
        S4.Rows.Add(-12.02673, -77.04)
        S4.Rows.Add(-12.02361, -77.03714)
        S4.Rows.Add(-12.02695, -77.03331)
```

```
        S4.Rows.Add(-12.03005, -77.03582)
        Dim Poligono_S4 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos4 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S4.Rows.Count - 1
            lat = Convert.ToDouble(S4(i)(0))
            lng = Convert.ToDouble(S4(i)(1))
            puntos4.Add(New PointLatLng(lat, lng))

        Next
        Dim poligonosPunto4 As GMapPolygon = New GMapPolygon(puntos4,
"Poligono")
        Poligono_S4.Polygons.Add(poligonosPunto4)
        Mapa.Overlays.Add(Poligono_S4)
        Dim S5 As DataTable
        S5 = New DataTable()
        S5.Columns.Add("")
        S5.Columns.Add("")
        S5.Rows.Add(-12.03178, -77.02771)
        S5.Rows.Add(-12.02729, -77.03292)
        S5.Rows.Add(-12.0245, -77.03032)
        S5.Rows.Add(-12.02736, -77.02741)
        S5.Rows.Add(-12.02766, -77.02742)
        S5.Rows.Add(-12.02797, -77.02754)
        S5.Rows.Add(-12.02777, -77.02734)
        S5.Rows.Add(-12.02768, -77.02698)
        S5.Rows.Add(-12.0277, -77.02668)
        S5.Rows.Add(-12.02946, -77.02445)
        S5.Rows.Add(-12.03178, -77.02771)
        Dim Poligono_S5 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos5 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S5.Rows.Count - 1
            lat = Convert.ToDouble(S5(i)(0))
            lng = Convert.ToDouble(S5(i)(1))
            puntos5.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto5 As GMapPolygon = New GMapPolygon(puntos5,
"Poligono")
        Poligono_S5.Polygons.Add(poligonosPunto5)
        Mapa.Overlays.Add(Poligono_S5)
        Dim S6 As DataTable
        S6 = New DataTable()
        S6.Columns.Add("")
        S6.Columns.Add("")
        S6.Rows.Add(-12.02729, -77.03292)
        S6.Rows.Add(-12.02695, -77.03331)
        S6.Rows.Add(-12.02361, -77.03714)
        S6.Rows.Add(-12.02238, -77.03607)
        S6.Rows.Add(-12.02293, -77.03543)
        S6.Rows.Add(-12.02281, -77.03343)
        S6.Rows.Add(-12.02173, -77.03244)
        S6.Rows.Add(-12.02282, -77.03103)
        S6.Rows.Add(-12.02333, -77.03113)
        S6.Rows.Add(-12.02385, -77.03101)
        S6.Rows.Add(-12.0245, -77.03032)
        S6.Rows.Add(-12.02729, -77.03292)
        Dim Poligono_S6 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos6 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S6.Rows.Count - 1
```

```
            lat = Convert.ToDouble(S6(i)(0))
            lng = Convert.ToDouble(S6(i)(1))
            puntos6.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto6 As GMapPolygon = New GMapPolygon(puntos6,
"Poligono")
        Poligono_S6.Polygons.Add(poligonosPunto6)
        Mapa.Overlays.Add(Poligono_S6)
        Mapa.Zoom = Mapa.Zoom + 1
        Mapa.Zoom = Mapa.Zoom - 1
    End Sub

    Sub Poligono_RI()
        Mapa.DragButton = MouseButtons.Left
        Mapa.CanDragMap = True
        Mapa.MapProvider = GMapProviders.GoogleMap
        Mapa.Position = New PointLatLng(-12.03867, -77.03238)
        Mapa.MinZoom = 0
        Mapa.MaxZoom = 24
        Mapa.Zoom = 15
        Mapa.AutoScroll = True
        'Agregando los poligonos del area de jurisdiccion
        Dim S1 As DataTable
        S1 = New DataTable()
        S1.Columns.Add("")
        S1.Columns.Add("")
        S1.Rows.Add(-12.03732, -77.04058)
        S1.Rows.Add(-12.03667, -77.04345)
        S1.Rows.Add(-12.02829, -77.04419)
        S1.Rows.Add(-12.0298, -77.04284)
        S1.Rows.Add(-12.03065, -77.04215)
        S1.Rows.Add(-12.03221, -77.04038)
        S1.Rows.Add(-12.03444, -77.03793)
        S1.Rows.Add(-12.03605, -77.03937)
        S1.Rows.Add(-12.03732, -77.04058)
        Dim Poligono_S1 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        Dim lat As Double
        Dim lng As Double
        For i = 0 To S1.Rows.Count - 1
            lat = Convert.ToDouble(S1(i)(0))
            lng = Convert.ToDouble(S1(i)(1))
            puntos.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto As GMapPolygon = New GMapPolygon(puntos,
"Poligono")
        Poligono_S1.Polygons.Add(poligonosPunto)
        Mapa.Overlays.Add(Poligono_S1)
        Dim S2 As DataTable
        S2 = New DataTable()
        S2.Columns.Add("")
        S2.Columns.Add("")
        S2.Rows.Add(-12.03667, -77.04345)
        S2.Rows.Add(-12.03773, -77.04331)
        S2.Rows.Add(-12.03839, -77.0421)
        S2.Rows.Add(-12.03862, -77.04008)
        S2.Rows.Add(-12.03905, -77.0382)
        S2.Rows.Add(-12.04092, -77.03371)
        S2.Rows.Add(-12.0387, -77.03227)
        S2.Rows.Add(-12.03444, -77.03793)
        S2.Rows.Add(-12.03605, -77.03937)
        S2.Rows.Add(-12.03732, -77.04058)
```

```
        S2.Rows.Add(-12.03667, -77.04345)
        Dim Poligono_S2 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos2 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S2.Rows.Count - 1
            lat = Convert.ToDouble(S2(i)(0))
            lng = Convert.ToDouble(S2(i)(1))
            puntos2.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto2 As GMapPolygon = New GMapPolygon(puntos2,
"Poligono")
        Poligono_S2.Polygons.Add(poligonosPunto2)
        Mapa.Overlays.Add(Poligono_S2)
        Dim S3 As DataTable
        S3 = New DataTable()
        S3.Columns.Add("")
        S3.Columns.Add("")
        S3.Rows.Add(-12.04092, -77.03371)
        S3.Rows.Add(-12.03871, -77.03228)
        S3.Rows.Add(-12.03983, -77.0309)
        S3.Rows.Add(-12.0401, -77.02837)
        S3.Rows.Add(-12.04034, -77.02253)
        S3.Rows.Add(-12.04197, -77.02257)
        S3.Rows.Add(-12.04226, -77.02386)
        S3.Rows.Add(-12.04252, -77.02383)
        S3.Rows.Add(-12.04255, -77.02394)
        S3.Rows.Add(-12.04339, -77.02436)
        S3.Rows.Add(-12.04303, -77.02906)
        S3.Rows.Add(-12.04092, -77.03371)
        Dim Poligono_S3 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos3 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S3.Rows.Count - 1
            lat = Convert.ToDouble(S3(i)(0))
            lng = Convert.ToDouble(S3(i)(1))
            puntos3.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto3 As GMapPolygon = New GMapPolygon(puntos3,
"Poligono")
        Poligono_S3.Polygons.Add(poligonosPunto3)
        Mapa.Overlays.Add(Poligono_S3)
        Dim S4 As DataTable
        S4 = New DataTable()
        S4.Columns.Add("")
        S4.Columns.Add("")
        S4.Rows.Add(-12.0401, -77.02837)
        S4.Rows.Add(-12.03983, -77.0309)
        S4.Rows.Add(-12.03871, -77.03228)
        S4.Rows.Add(-12.03201, -77.02776)
        S4.Rows.Add(-12.03482, -77.02516)
        S4.Rows.Add(-12.03783, -77.02683)
        S4.Rows.Add(-12.03795, -77.0271)
        S4.Rows.Add(-12.0401, -77.02837)
        Dim Poligono_S4 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos4 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S4.Rows.Count - 1
            lat = Convert.ToDouble(S4(i)(0))
            lng = Convert.ToDouble(S4(i)(1))
            puntos4.Add(New PointLatLng(lat, lng))
        Next
```

```
        Dim poligonosPunto4 As GMapPolygon = New GMapPolygon(puntos4,
"Poligono")
        Poligono_S4.Polygons.Add(poligonosPunto4)
        Mapa.Overlays.Add(Poligono_S4)
        Dim S5 As DataTable
        S5 = New DataTable()
        S5.Columns.Add("")
        S5.Columns.Add("")
        S5.Rows.Add(-12.04034, -77.02253)
        S5.Rows.Add(-12.03903, -77.02298)
        S5.Rows.Add(-12.03876, -77.02232)
        S5.Rows.Add(-12.03714, -77.02148)
        S5.Rows.Add(-12.03669, -77.02154)
        S5.Rows.Add(-12.03641, -77.02135)
        S5.Rows.Add(-12.03605, -77.02124)
        S5.Rows.Add(-12.03543, -77.02072)
        S5.Rows.Add(-12.03484, -77.02059)
        S5.Rows.Add(-12.03453, -77.02077)
        S5.Rows.Add(-12.03425, -77.02123)
        S5.Rows.Add(-12.03425, -77.02216)
        S5.Rows.Add(-12.03453, -77.02339)
        S5.Rows.Add(-12.03437, -77.02485)
        S5.Rows.Add(-12.03482, -77.02516)
        S5.Rows.Add(-12.03783, -77.02683)
        S5.Rows.Add(-12.03795, -77.0271)
        S5.Rows.Add(-12.0401, -77.02837)
        S5.Rows.Add(-12.04034, -77.02253)
        Dim Poligono_S5 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos5 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S5.Rows.Count - 1
            lat = Convert.ToDouble(S5(i)(0))
            lng = Convert.ToDouble(S5(i)(1))
            puntos5.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto5 As GMapPolygon = New GMapPolygon(puntos5,
"Poligono")
        Poligono_S5.Polygons.Add(poligonosPunto5)
        Mapa.Overlays.Add(Poligono_S5)
        Dim S6 As DataTable
        S6 = New DataTable()
        S6.Columns.Add("")
        S6.Columns.Add("")
        S6.Rows.Add(-12.03482, -77.02516)
        S6.Rows.Add(-12.03201, -77.02776)
        S6.Rows.Add(-12.03192, -77.02768)
        S6.Rows.Add(-12.03184, -77.02778)
        S6.Rows.Add(-12.03179, -77.02771)
        S6.Rows.Add(-12.03119, -77.02803)
        S6.Rows.Add(-12.02906, -77.02534)
        S6.Rows.Add(-12.02939, -77.02458)
        S6.Rows.Add(-12.02923, -77.02388)
        S6.Rows.Add(-12.02955, -77.02307)
        S6.Rows.Add(-12.03036, -77.02237)
        S6.Rows.Add(-12.03186, -77.02176)
        S6.Rows.Add(-12.03232, -77.02133)
        S6.Rows.Add(-12.03365, -77.02086)
        S6.Rows.Add(-12.03484, -77.02059)
        S6.Rows.Add(-12.03453, -77.02077)
        S6.Rows.Add(-12.03425, -77.02123)
        S6.Rows.Add(-12.03425, -77.02216)
        S6.Rows.Add(-12.03453, -77.02339)
        S6.Rows.Add(-12.03437, -77.02485)
```

```
        S6.Rows.Add(-12.03482, -77.02516)
        Dim Poligono_S6 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos6 As List(Of PointLatLng) = New List(Of PointLatLng) '---> 
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S6.Rows.Count - 1
            lat = Convert.ToDouble(S6(i)(0))
            lng = Convert.ToDouble(S6(i)(1))
            puntos6.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto6 As GMapPolygon = New GMapPolygon(puntos6, 
"Poligono")
        Poligono_S6.Polygons.Add(poligonosPunto6)
        Mapa.Overlays.Add(Poligono_S6)
        Mapa.Zoom = Mapa.Zoom + 1
        Mapa.Zoom = Mapa.Zoom - 1
    End Sub
    Sub Poligono_PD()
        Mapa.DragButton = MouseButtons.Left
        Mapa.CanDragMap = True
        Mapa.MapProvider = GMapProviders.GoogleMap
        Mapa.Position = New PointLatLng(-12.03876, -77.0197)
        Mapa.MinZoom = 0
        Mapa.MaxZoom = 24
        Mapa.Zoom = 15
        Mapa.AutoScroll = True
        'Agregando mapa con colores
        Dim S1_1 As DataTable
        S1_1 = New DataTable()
        S1_1.Columns.Add("")
        S1_1.Columns.Add("")
        S1_1.Rows.Add(-12.04187, -77.0226)
        S1_1.Rows.Add(-12.04151, -77.02255)
        S1_1.Rows.Add(-12.04113, -77.02255)
        S1_1.Rows.Add(-12.0404, -77.02257)
        S1_1.Rows.Add(-12.04035, -77.02275)
        S1_1.Rows.Add(-12.03902, -77.02292)
        S1_1.Rows.Add(-12.03897, -77.0226)
        S1_1.Rows.Add(-12.03869, -77.02228)
        S1_1.Rows.Add(-12.0381, -77.02195)
        S1_1.Rows.Add(-12.03759, -77.02183)
        S1_1.Rows.Add(-12.03722, -77.02155)
        S1_1.Rows.Add(-12.03698, -77.02154)
        S1_1.Rows.Add(-12.0367, -77.02156)
        S1_1.Rows.Add(-12.03641, -77.02138)
        S1_1.Rows.Add(-12.03608, -77.02131)
        S1_1.Rows.Add(-12.03601, -77.02127)
        S1_1.Rows.Add(-12.0365, -77.0208)
        S1_1.Rows.Add(-12.0377, -77.02041)
        S1_1.Rows.Add(-12.03768, -77.01997)
        S1_1.Rows.Add(-12.03786, -77.01929)
        S1_1.Rows.Add(-12.03831, -77.01897)
        S1_1.Rows.Add(-12.03833, -77.01861)
        S1_1.Rows.Add(-12.03818, -77.01823)
        S1_1.Rows.Add(-12.03856, -77.01782)
        S1_1.Rows.Add(-12.0389, -77.01784)
        S1_1.Rows.Add(-12.03925, -77.01761)
        S1_1.Rows.Add(-12.03938, -77.01799)
        S1_1.Rows.Add(-12.04009, -77.01808)
        S1_1.Rows.Add(-12.04039, -77.01783)
        S1_1.Rows.Add(-12.04054, -77.01816)
        S1_1.Rows.Add(-12.04057, -77.01846)
        S1_1.Rows.Add(-12.04085, -77.01935)
        S1_1.Rows.Add(-12.04119, -77.02036)
```

```
        S1_1.Rows.Add(-12.04161, -77.02163)
        S1_1.Rows.Add(-12.04187, -77.0226)
        Dim Poligono_S1_1 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        Dim lat As Double
        Dim lng As Double
        For i = 0 To S1_1.Rows.Count - 1
            lat = Convert.ToDouble(S1_1(i)(0))
            lng = Convert.ToDouble(S1_1(i)(1))
            puntos.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto As GMapPolygon = New GMapPolygon(puntos,
"Poligono")
        Poligono_S1_1.Polygons.Add(poligonosPunto)
        Mapa.Overlays.Add(Poligono_S1_1)
        Dim S1_2 As DataTable
        S1_2 = New DataTable()
        S1_2.Columns.Add("")
        S1_2.Columns.Add("")
        S1_2.Rows.Add(-12.04039, -77.01783)
        S1_2.Rows.Add(-12.04009, -77.01808)
        S1_2.Rows.Add(-12.03938, -77.01799)
        S1_2.Rows.Add(-12.03925, -77.01761)
        S1_2.Rows.Add(-12.0389, -77.01784)
        S1_2.Rows.Add(-12.03856, -77.01782)
        S1_2.Rows.Add(-12.03818, -77.01823)
        S1_2.Rows.Add(-12.03833, -77.01861)
        S1_2.Rows.Add(-12.03831, -77.01897)
        S1_2.Rows.Add(-12.03786, -77.01929)
        S1_2.Rows.Add(-12.03768, -77.01997)
        S1_2.Rows.Add(-12.0377, -77.02041)
        S1_2.Rows.Add(-12.0365, -77.0208)
        S1_2.Rows.Add(-12.03601, -77.02127)
        S1_2.Rows.Add(-12.03561, -77.02087)
        S1_2.Rows.Add(-12.03527, -77.02065)
        S1_2.Rows.Add(-12.03496, -77.0206)
        S1_2.Rows.Add(-12.0347, -77.02068)
        S1_2.Rows.Add(-12.03459, -77.02077)
        S1_2.Rows.Add(-12.03441, -77.02092)
        S1_2.Rows.Add(-12.03435, -77.02114)
        S1_2.Rows.Add(-12.03421, -77.02123)
        S1_2.Rows.Add(-12.03382, -77.02062)
        S1_2.Rows.Add(-12.03354, -77.01999)
        S1_2.Rows.Add(-12.03333, -77.01987)
        S1_2.Rows.Add(-12.033, -77.01987)
        S1_2.Rows.Add(-12.03292, -77.01965)
        S1_2.Rows.Add(-12.03207, -77.01908)
        S1_2.Rows.Add(-12.03243, -77.01853)
        S1_2.Rows.Add(-12.03249, -77.01824)
        S1_2.Rows.Add(-12.03277, -77.01779)
        S1_2.Rows.Add(-12.03371, -77.01765)
        S1_2.Rows.Add(-12.03415, -77.0178)
        S1_2.Rows.Add(-12.03389, -77.01734)
        S1_2.Rows.Add(-12.03381, -77.01707)
        S1_2.Rows.Add(-12.03383, -77.01687)
        S1_2.Rows.Add(-12.03396, -77.01664)
        S1_2.Rows.Add(-12.03468, -77.01618)
        S1_2.Rows.Add(-12.03497, -77.01609)
        S1_2.Rows.Add(-12.03562, -77.0157)
        S1_2.Rows.Add(-12.03625, -77.01565)
        S1_2.Rows.Add(-12.03648, -77.01555)
        S1_2.Rows.Add(-12.03661, -77.01538)
```

```
        S1_2.Rows.Add(-12.03664, -77.01516)
        S1_2.Rows.Add(-12.03693, -77.01482)
        S1_2.Rows.Add(-12.0371, -77.01471)
        S1_2.Rows.Add(-12.03696, -77.01449)
        S1_2.Rows.Add(-12.03717, -77.01416)
        S1_2.Rows.Add(-12.03724, -77.01362)
        S1_2.Rows.Add(-12.03742, -77.01327)
        S1_2.Rows.Add(-12.03849, -77.01519)
        S1_2.Rows.Add(-12.03956, -77.01655)
        S1_2.Rows.Add(-12.04039, -77.01783)
        Dim Poligono_S1_2 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos2 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S1_2.Rows.Count - 1
            lat = Convert.ToDouble(S1_2(i)(0))
            lng = Convert.ToDouble(S1_2(i)(1))
            puntos2.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto2 As GMapPolygon = New GMapPolygon(puntos2,
"Poligono")
        Poligono_S1_2.Polygons.Add(poligonosPunto2)
        Mapa.Overlays.Add(Poligono_S1_2)
        'Actualizar el mapa
        Mapa.Zoom = Mapa.Zoom + 1
        Mapa.Zoom = Mapa.Zoom - 1
        Dim S2_1 As DataTable
        S2_1 = New DataTable()
        S2_1.Columns.Add("")
        S2_1.Columns.Add("")
        S2_1.Rows.Add(-12.04187, -77.0226)
        S2_1.Rows.Add(-12.0422, -77.02388)
        S2_1.Rows.Add(-12.04252, -77.02384)
        S2_1.Rows.Add(-12.04277, -77.02401)
        S2_1.Rows.Add(-12.04346, -77.02435)
        S2_1.Rows.Add(-12.04352, -77.023)
        S2_1.Rows.Add(-12.04328, -77.02125)
        S2_1.Rows.Add(-12.04357, -77.01929)
        S2_1.Rows.Add(-12.0422, -77.01982)
        S2_1.Rows.Add(-12.04119, -77.02036)
        S2_1.Rows.Add(-12.04161, -77.02163)
        S2_1.Rows.Add(-12.04187, -77.0226)
        Dim Poligono_S2_1 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos3 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S2_1.Rows.Count - 1
            lat = Convert.ToDouble(S2_1(i)(0))
            lng = Convert.ToDouble(S2_1(i)(1))
            puntos3.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto3 As GMapPolygon = New GMapPolygon(puntos3,
"Poligono")
        Poligono_S2_1.Polygons.Add(poligonosPunto3)
        Mapa.Overlays.Add(Poligono_S2_1)
        Dim S2_2 As DataTable
        S2_2 = New DataTable()
        S2_2.Columns.Add("")
        S2_2.Columns.Add("")
        S2_2.Rows.Add(-12.04119, -77.02036)
        S2_2.Rows.Add(-12.04085, -77.01935)
        S2_2.Rows.Add(-12.04057, -77.01846)
        S2_2.Rows.Add(-12.04054, -77.01816)
        S2_2.Rows.Add(-12.04039, -77.01783)
```

```
        S2_2.Rows.Add(-12.03956, -77.01655)
        S2_2.Rows.Add(-12.03849, -77.01519)
        S2_2.Rows.Add(-12.03742, -77.01327)
        S2_2.Rows.Add(-12.03784, -77.01332)
        S2_2.Rows.Add(-12.03899, -77.01384)
        S2_2.Rows.Add(-12.04085, -77.01429)
        S2_2.Rows.Add(-12.04172, -77.01504)
        S2_2.Rows.Add(-12.04318, -77.01697)
        S2_2.Rows.Add(-12.04353, -77.01797)
        S2_2.Rows.Add(-12.04357, -77.01929)
        S2_2.Rows.Add(-12.0422, -77.01982)
        S2_2.Rows.Add(-12.04119, -77.02036)
        Dim Poligono_S2_2 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos4 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S2_2.Rows.Count - 1
            lat = Convert.ToDouble(S2_2(i)(0))
            lng = Convert.ToDouble(S2_2(i)(1))
            puntos4.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto4 As GMapPolygon = New GMapPolygon(puntos4,
"Poligono")
        Poligono_S2_2.Polygons.Add(poligonosPunto4)
        Mapa.Overlays.Add(Poligono_S2_2)
        Mapa.Zoom = Mapa.Zoom + 1
        Mapa.Zoom = Mapa.Zoom - 1
    End Sub
    Sub Poligono_CC()
        Mapa.DragButton = MouseButtons.Left
        Mapa.CanDragMap = True
        Mapa.MapProvider = GMapProviders.GoogleMap
        Mapa.Position = New PointLatLng(-12.01921, -77.03346)
        Mapa.MinZoom = 0
        Mapa.MaxZoom = 24
        Mapa.Zoom = 15
        Mapa.AutoScroll = True
        'Agregando los poligonos del area de jurisdiccion
        Dim S1 As DataTable
        S1 = New DataTable()
        S1.Columns.Add("")
        S1.Columns.Add("")
        S1.Rows.Add(-12.0299, -77.04297)
        S1.Rows.Add(-12.02264, -77.0504)
        S1.Rows.Add(-12.01092, -77.05281)
        S1.Rows.Add(-12.01051, -77.05035)
        S1.Rows.Add(-12.01091, -77.0496)
        S1.Rows.Add(-12.01183, -77.04925)
        S1.Rows.Add(-12.01238, -77.04929)
        S1.Rows.Add(-12.01273, -77.04879)
        S1.Rows.Add(-12.01382, -77.04905)
        S1.Rows.Add(-12.0145, -77.0488)
        S1.Rows.Add(-12.01471, -77.04902)
        S1.Rows.Add(-12.01546, -77.04847)
        S1.Rows.Add(-12.01548, -77.04803)
        S1.Rows.Add(-12.01573, -77.0475)
        S1.Rows.Add(-12.01621, -77.04715)
        S1.Rows.Add(-12.016, -77.04634)
        S1.Rows.Add(-12.0164, -77.04542)
        S1.Rows.Add(-12.01898, -77.0452)
        S1.Rows.Add(-12.02102, -77.04544)
        S1.Rows.Add(-12.02364, -77.04404)
        S1.Rows.Add(-12.02379, -77.04318)
        S1.Rows.Add(-12.0232, -77.04271)
```

```
        S1.Rows.Add(-12.02203, -77.04291)
        S1.Rows.Add(-12.02058, -77.04271)
        S1.Rows.Add(-12.01995, -77.04237)
        S1.Rows.Add(-12.0193, -77.04134)
        S1.Rows.Add(-12.01869, -77.04033)
        S1.Rows.Add(-12.0198, -77.0363)
        S1.Rows.Add(-12.019, -77.03629)
        S1.Rows.Add(-12.01944, -77.03588)
        S1.Rows.Add(-12.01864, -77.03503)
        S1.Rows.Add(-12.02242, -77.03081)
        S1.Rows.Add(-12.02287, -77.03108)
        S1.Rows.Add(-12.02183, -77.03246)
        S1.Rows.Add(-12.02285, -77.03344)
        S1.Rows.Add(-12.02292, -77.03535)
        S1.Rows.Add(-12.02237, -77.03607)
        S1.Rows.Add(-12.02346, -77.03708)
        S1.Rows.Add(-12.02368, -77.03705)
        S1.Rows.Add(-12.0299, -77.04297)
        Dim Poligono_S1 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        Dim lat As Double
        Dim lng As Double
        For i = 0 To S1.Rows.Count - 1
            lat = Convert.ToDouble(S1(i)(0))
            lng = Convert.ToDouble(S1(i)(1))
            puntos.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto As GMapPolygon = New GMapPolygon(puntos,
"Poligono")
        Poligono_S1.Polygons.Add(poligonosPunto)
        Mapa.Overlays.Add(Poligono_S1)
        Dim S2 As DataTable
        S2 = New DataTable()
        S2.Columns.Add("")
        S2.Columns.Add("")
        S2.Rows.Add(-12.02242, -77.03081)
        S2.Rows.Add(-12.01864, -77.03503)
        S2.Rows.Add(-12.01944, -77.03588)
        S2.Rows.Add(-12.019, -77.03629)
        S2.Rows.Add(-12.01848, -77.03617)
        S2.Rows.Add(-12.01774, -77.03573)
        S2.Rows.Add(-12.01684, -77.0354)
        S2.Rows.Add(-12.01631, -77.03493)
        S2.Rows.Add(-12.01611, -77.03405)
        S2.Rows.Add(-12.01593, -77.03376)
        S2.Rows.Add(-12.01606, -77.03311)
        S2.Rows.Add(-12.01654, -77.03309)
        S2.Rows.Add(-12.01694, -77.03247)
        S2.Rows.Add(-12.01799, -77.03192)
        S2.Rows.Add(-12.0173, -77.03115)
        S2.Rows.Add(-12.02018, -77.02849)
        S2.Rows.Add(-12.02242, -77.03081)
        Dim Poligono_S2 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos2 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S2.Rows.Count - 1
            lat = Convert.ToDouble(S2(i)(0))
            lng = Convert.ToDouble(S2(i)(1))
            puntos2.Add(New PointLatLng(lat, lng))
        Next
```

```
        Dim poligonosPunto2 As GMapPolygon = New GMapPolygon(puntos2,
"Poligono")
        Poligono_S2.Polygons.Add(poligonosPunto2)
        Mapa.Overlays.Add(Poligono_S2)
        Dim S3 As DataTable
        S3 = New DataTable()
        S3.Columns.Add("")
        S3.Columns.Add("")
        S3.Rows.Add(-12.01946, -77.02835)
        S3.Rows.Add(-12.02018, -77.02849)
        S3.Rows.Add(-12.02242, -77.03081)
        S3.Rows.Add(-12.02287, -77.03108)
        S3.Rows.Add(-12.02359, -77.03111)
        S3.Rows.Add(-12.02402, -77.03084)
        S3.Rows.Add(-12.02741, -77.02744)
        S3.Rows.Add(-12.02721, -77.02716)
        S3.Rows.Add(-12.02664, -77.0271)
        S3.Rows.Add(-12.02664, -77.0268)
        S3.Rows.Add(-12.02639, -77.02657)
        S3.Rows.Add(-12.02603, -77.02529)
        S3.Rows.Add(-12.02586, -77.02514)
        S3.Rows.Add(-12.02559, -77.02514)
        S3.Rows.Add(-12.02499, -77.02549)
        S3.Rows.Add(-12.02274, -77.02792)
        S3.Rows.Add(-12.02155, -77.02672)
        S3.Rows.Add(-12.02116, -77.02716)
        S3.Rows.Add(-12.02125, -77.02764)
        S3.Rows.Add(-12.0207, -77.02817)
        S3.Rows.Add(-12.0205, -77.02813)
        S3.Rows.Add(-12.02008, -77.02792)
        S3.Rows.Add(-12.01963, -77.02808)
        S3.Rows.Add(-12.01946, -77.02835)
        Dim Poligono_S3 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos3 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S3.Rows.Count - 1
            lat = Convert.ToDouble(S3(i)(0))
            lng = Convert.ToDouble(S3(i)(1))
            puntos3.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto3 As GMapPolygon = New GMapPolygon(puntos3,
"Poligono")
        Poligono_S3.Polygons.Add(poligonosPunto3)
        Mapa.Overlays.Add(Poligono_S3)
        Dim S4 As DataTable
        S4 = New DataTable()
        S4.Columns.Add("")
        S4.Columns.Add("")
        S4.Rows.Add(-12.01795, -77.02237)
        S4.Rows.Add(-12.01867, -77.0229)
        S4.Rows.Add(-12.01929, -77.02361)
        S4.Rows.Add(-12.01959, -77.02444)
        S4.Rows.Add(-12.01973, -77.02544)
        S4.Rows.Add(-12.01969, -77.02729)
        S4.Rows.Add(-12.01946, -77.02835)
        S4.Rows.Add(-12.01963, -77.02808)
        S4.Rows.Add(-12.02008, -77.02792)
        S4.Rows.Add(-12.0205, -77.02813)
        S4.Rows.Add(-12.0207, -77.02817)
        S4.Rows.Add(-12.02125, -77.02764)
        S4.Rows.Add(-12.02116, -77.02716)
        S4.Rows.Add(-12.02155, -77.02672)
        S4.Rows.Add(-12.02274, -77.02792)
```

```
        S4.Rows.Add(-12.02499, -77.02549)
        S4.Rows.Add(-12.02559, -77.02514)
        S4.Rows.Add(-12.02586, -77.02514)
        S4.Rows.Add(-12.02603, -77.02529)
        S4.Rows.Add(-12.02639, -77.02657)
        S4.Rows.Add(-12.02664, -77.0268)
        S4.Rows.Add(-12.02664, -77.0271)
        S4.Rows.Add(-12.02721, -77.02716)
        S4.Rows.Add(-12.02741, -77.02744)
        S4.Rows.Add(-12.02798, -77.02754)
        S4.Rows.Add(-12.02771, -77.02707)
        S4.Rows.Add(-12.02777, -77.0263)
        S4.Rows.Add(-12.02601, -77.02456)
        S4.Rows.Add(-12.02452, -77.02385)
        S4.Rows.Add(-12.02404, -77.02291)
        S4.Rows.Add(-12.02349, -77.0225)
        S4.Rows.Add(-12.02169, -77.02243)
        S4.Rows.Add(-12.02093, -77.02201)
        S4.Rows.Add(-12.02036, -77.0219)
        S4.Rows.Add(-12.01795, -77.02237)
        Dim Poligono_S4 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos4 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S4.Rows.Count - 1
            lat = Convert.ToDouble(S4(i)(0))
            lng = Convert.ToDouble(S4(i)(1))
            puntos4.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto4 As GMapPolygon = New GMapPolygon(puntos4,
"Poligono")
        Poligono_S4.Polygons.Add(poligonosPunto4)
        Mapa.Overlays.Add(Poligono_S4)
        Dim S5 As DataTable
        S5 = New DataTable()
        S5.Columns.Add("")
        S5.Columns.Add("")
        S5.Rows.Add(-12.01959, -77.02444)
        S5.Rows.Add(-12.01973, -77.02544)
        S5.Rows.Add(-12.01969, -77.02729)
        S5.Rows.Add(-12.01946, -77.02835)
        S5.Rows.Add(-12.02018, -77.02849)
        S5.Rows.Add(-12.0173, -77.03115)
        S5.Rows.Add(-12.01668, -77.03048)
        S5.Rows.Add(-12.0158, -77.02823)
        S5.Rows.Add(-12.01539, -77.02853)
        S5.Rows.Add(-12.0146, -77.02776)
        S5.Rows.Add(-12.01339, -77.0276)
        S5.Rows.Add(-12.01262, -77.02693)
        S5.Rows.Add(-12.01348, -77.02664)
        S5.Rows.Add(-12.01369, -77.02639)
        S5.Rows.Add(-12.01455, -77.02597)
        S5.Rows.Add(-12.01565, -77.02523)
        S5.Rows.Add(-12.01623, -77.02431)
        S5.Rows.Add(-12.01656, -77.02461)
        S5.Rows.Add(-12.01714, -77.02454)
        S5.Rows.Add(-12.01788, -77.02485)
        S5.Rows.Add(-12.01826, -77.02425)
        S5.Rows.Add(-12.01959, -77.02444)
        Dim Poligono_S5 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos5 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S5.Rows.Count - 1
```

```
            lat = Convert.ToDouble(S5(i)(0))
            lng = Convert.ToDouble(S5(i)(1))
            puntos5.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto5 As GMapPolygon = New GMapPolygon(puntos5,
"Poligono")
        Poligono_S5.Polygons.Add(poligonosPunto5)
        Mapa.Overlays.Add(Poligono_S5)
        Dim S6 As DataTable
        S6 = New DataTable()
        S6.Columns.Add("")
        S6.Columns.Add("")
        S6.Rows.Add(-12.01065, -77.02235)
        S6.Rows.Add(-12.01104, -77.02405)
        S6.Rows.Add(-12.01183, -77.02596)
        S6.Rows.Add(-12.01262, -77.02693)
        S6.Rows.Add(-12.01348, -77.02664)
        S6.Rows.Add(-12.01369, -77.02639)
        S6.Rows.Add(-12.01455, -77.02597)
        S6.Rows.Add(-12.01565, -77.02523)
        S6.Rows.Add(-12.01623, -77.02431)
        S6.Rows.Add(-12.01656, -77.02461)
        S6.Rows.Add(-12.01714, -77.02454)
        S6.Rows.Add(-12.01788, -77.02485)
        S6.Rows.Add(-12.01826, -77.02425)
        S6.Rows.Add(-12.01959, -77.02444)
        S6.Rows.Add(-12.01929, -77.02361)
        S6.Rows.Add(-12.01867, -77.0229)
        S6.Rows.Add(-12.01795, -77.02237)
        S6.Rows.Add(-12.01585, -77.02235)
        S6.Rows.Add(-12.0153, -77.02253)
        S6.Rows.Add(-12.01457, -77.02242)
        S6.Rows.Add(-12.01392, -77.02246)
        S6.Rows.Add(-12.01294, -77.02209)
        S6.Rows.Add(-12.01065, -77.02235)
        Dim Poligono_S6 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos6 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S6.Rows.Count - 1
            lat = Convert.ToDouble(S6(i)(0))
            lng = Convert.ToDouble(S6(i)(1))
            puntos6.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto6 As GMapPolygon = New GMapPolygon(puntos6,
"Poligono")
        Poligono_S6.Polygons.Add(poligonosPunto6)
        Mapa.Overlays.Add(Poligono_S6)
        Mapa.Zoom = Mapa.Zoom + 1
        Mapa.Zoom = Mapa.Zoom - 1
    End Sub
    Sub Poligono_FA()
        Mapa.DragButton = MouseButtons.Left
        Mapa.CanDragMap = True
        Mapa.MapProvider = GMapProviders.GoogleMap
        Mapa.Position = New PointLatLng(-12.01076, -77.03321)
        Mapa.MinZoom = 0
        Mapa.MaxZoom = 24
        Mapa.Zoom = 15
        Mapa.AutoScroll = True
        'Agregando los poligonos del area de jurisdiccion
        Dim S1 As DataTable
        S1 = New DataTable()
        S1.Columns.Add("")
```

```
        S1.Columns.Add("")
        S1.Rows.Add(-11.99974, -77.03357)
        S1.Rows.Add(-12.00048, -77.03415)
        S1.Rows.Add(-12.00176, -77.03484)
        S1.Rows.Add(-12.00297, -77.03593)
        S1.Rows.Add(-12.00375, -77.03683)
        S1.Rows.Add(-12.00436, -77.0381)
        S1.Rows.Add(-12.00499, -77.03868)
        S1.Rows.Add(-12.00709, -77.03817)
        S1.Rows.Add(-12.00768, -77.03775)
        S1.Rows.Add(-12.0081, -77.03861)
        S1.Rows.Add(-12.00873, -77.03837)
        S1.Rows.Add(-12.00823, -77.03657)
        S1.Rows.Add(-12.00779, -77.03682)
        S1.Rows.Add(-12.00581, -77.03305)
        S1.Rows.Add(-12.00508, -77.03292)
        S1.Rows.Add(-12.00485, -77.03202)
        S1.Rows.Add(-12.00426, -77.03142)
        S1.Rows.Add(-12.0039, -77.03064)
        S1.Rows.Add(-12.00195, -77.03179)
        S1.Rows.Add(-12.00018, -77.03302)
        S1.Rows.Add(-11.99974, -77.03357)
        Dim Poligono_S1 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        Dim lat As Double
        Dim lng As Double
        For i = 0 To S1.Rows.Count - 1
            lat = Convert.ToDouble(S1(i)(0))
            lng = Convert.ToDouble(S1(i)(1))
            puntos.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto As GMapPolygon = New GMapPolygon(puntos,
"Poligono")
        Poligono_S1.Polygons.Add(poligonosPunto)
        Mapa.Overlays.Add(Poligono_S1)
        Dim S2 As DataTable
        S2 = New DataTable()
        S2.Columns.Add("")
        S2.Columns.Add("")
        S2.Rows.Add(-12.01319, -77.03716)
        S2.Rows.Add(-12.01209, -77.03725)
        S2.Rows.Add(-12.0118, -77.03678)
        S2.Rows.Add(-12.01003, -77.03803)
        S2.Rows.Add(-12.00873, -77.03837)
        S2.Rows.Add(-12.00823, -77.03657)
        S2.Rows.Add(-12.00779, -77.03682)
        S2.Rows.Add(-12.00581, -77.03305)
        S2.Rows.Add(-12.00508, -77.03292)
        S2.Rows.Add(-12.00485, -77.03202)
        S2.Rows.Add(-12.00426, -77.03142)
        S2.Rows.Add(-12.0039, -77.03064)
        S2.Rows.Add(-12.00546, -77.02991)
        S2.Rows.Add(-12.00747, -77.02734)
        S2.Rows.Add(-12.00898, -77.02807)
        S2.Rows.Add(-12.00995, -77.02811)
        S2.Rows.Add(-12.01201, -77.02871)
        S2.Rows.Add(-12.01226, -77.02859)
        S2.Rows.Add(-12.0123, -77.02932)
        S2.Rows.Add(-12.01264, -77.03031)
        S2.Rows.Add(-12.01231, -77.03088)
        S2.Rows.Add(-12.01228, -77.03155)
        S2.Rows.Add(-12.01263, -77.03188)
```

```
        S2.Rows.Add(-12.0131, -77.03275)
        S2.Rows.Add(-12.01291, -77.03672)
        S2.Rows.Add(-12.01319, -77.03716)
        Dim Poligono_S2 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos2 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S2.Rows.Count - 1
            lat = Convert.ToDouble(S2(i)(0))
            lng = Convert.ToDouble(S2(i)(1))
            puntos2.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto2 As GMapPolygon = New GMapPolygon(puntos2,
"Poligono")
        Poligono_S2.Polygons.Add(poligonosPunto2)
        Mapa.Overlays.Add(Poligono_S2)
        Dim S3 As DataTable
        S3 = New DataTable()
        S3.Columns.Add("")
        S3.Columns.Add("")
        S3.Rows.Add(-12.01814, -77.03565)
        S3.Rows.Add(-12.01709, -77.03589)
        S3.Rows.Add(-12.016, -77.03653)
        S3.Rows.Add(-12.01587, -77.03685)
        S3.Rows.Add(-12.01487, -77.03744)
        S3.Rows.Add(-12.01436, -77.03736)
        S3.Rows.Add(-12.01352, -77.03738)
        S3.Rows.Add(-12.01319, -77.03716)
        S3.Rows.Add(-12.01291, -77.03672)
        S3.Rows.Add(-12.0131, -77.03275)
        S3.Rows.Add(-12.01511, -77.03365)
        S3.Rows.Add(-12.01658, -77.03307)
        S3.Rows.Add(-12.01702, -77.03237)
        S3.Rows.Add(-12.01795, -77.03187)
        S3.Rows.Add(-12.01827, -77.03234)
        S3.Rows.Add(-12.01778, -77.03256)
        S3.Rows.Add(-12.01734, -77.03303)
        S3.Rows.Add(-12.01661, -77.03357)
        S3.Rows.Add(-12.01652, -77.03443)
        S3.Rows.Add(-12.01684, -77.0349)
        S3.Rows.Add(-12.0173, -77.03524)
        S3.Rows.Add(-12.01814, -77.03565)
        Dim Poligono_S3 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos3 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S3.Rows.Count - 1
            lat = Convert.ToDouble(S3(i)(0))
            lng = Convert.ToDouble(S3(i)(1))
            puntos3.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto3 As GMapPolygon = New GMapPolygon(puntos3,
"Poligono")
        Poligono_S3.Polygons.Add(poligonosPunto3)
        Mapa.Overlays.Add(Poligono_S3)
        Dim S4 As DataTable
        S4 = New DataTable()
        S4.Columns.Add("")
        S4.Columns.Add("")
        S4.Rows.Add(-12.01795, -77.03187)
        S4.Rows.Add(-12.01702, -77.03237)
        S4.Rows.Add(-12.01658, -77.03307)
        S4.Rows.Add(-12.01511, -77.03365)
        S4.Rows.Add(-12.0131, -77.03275)
```

```
        S4.Rows.Add(-12.01263, -77.03188)
        S4.Rows.Add(-12.01228, -77.03155)
        S4.Rows.Add(-12.01231, -77.03088)
        S4.Rows.Add(-12.01264, -77.03031)
        S4.Rows.Add(-12.0123, -77.02932)
        S4.Rows.Add(-12.01226, -77.02859)
        S4.Rows.Add(-12.01251, -77.02834)
        S4.Rows.Add(-12.01358, -77.02834)
        S4.Rows.Add(-12.01398, -77.02872)
        S4.Rows.Add(-12.01425, -77.02857)
        S4.Rows.Add(-12.01469, -77.02926)
        S4.Rows.Add(-12.01596, -77.02864)
        S4.Rows.Add(-12.01643, -77.02995)
        S4.Rows.Add(-12.01668, -77.03046)
        S4.Rows.Add(-12.01795, -77.03187)
        Dim Poligono_S4 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos4 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S4.Rows.Count - 1
            lat = Convert.ToDouble(S4(i)(0))
            lng = Convert.ToDouble(S4(i)(1))
            puntos4.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto4 As GMapPolygon = New GMapPolygon(puntos4,
"Poligono")
        Poligono_S4.Polygons.Add(poligonosPunto4)
        Mapa.Overlays.Add(Poligono_S4)
        Mapa.Zoom = Mapa.Zoom + 1
        Mapa.Zoom = Mapa.Zoom - 1
    End Sub
    Private Sub BEjecutar_Click(sender As Object, e As EventArgs) Handles
BEjecutar.Click
        Dim punto_inicial As String
        Dim fil_punto_inicial As Integer
        For i = 0 To PTactico.RowCount - 1
            If PTactico(1, i).Value = Enabled Then
                punto_inicial = PTactico(0, i).Value
                fil_punto_inicial = i
                Exit For
            End If
        Next
        punto_inicial = punto_inicial
        fil_punto_inicial = fil_punto_inicial
        '''''''''''''''''''''''''''''''''''''''''''''''''
        ' Construir la matrices del Resultados Finales
        Dim NumMaximoPuntos As Integer = NumPuntos.Text
        ResultadosFinales.DataGridView1.ColumnCount = NumMaximoPuntos + 2
        ResultadosFinales.DataGridView3.ColumnCount = NumMaximoPuntos + 2
        ResultadosFinales.DataGridView5.ColumnCount = NumMaximoPuntos + 2
        ResultadosFinales.DataGridView7.ColumnCount = NumMaximoPuntos + 2
        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        ' Construir la matriz de distancias euclidianas

        'Cuantos CheckBox activados
        Dim Num_acti As Integer = 0
        For f = 0 To PCriticos.RowCount - 1
            If PCriticos(1, f).Value = Enabled Then
                Num_acti = Num_acti + 1
            End If
        Next
        Num_acti = Num_acti
        '''''''''''''''''''''''''''''''''''''''''''
        'Las dimensiones de la matriz
```

```
        Dim tabla_matriz As DataTable
        tabla_matriz = New DataTable()
        'Agregando dos columnas y dos filas
        tabla_matriz.Columns.Add()
        tabla_matriz.Rows.Add()
        tabla_matriz.Columns.Add()
        tabla_matriz.Rows.Add()
        '''''''''''''''''''''''''''''''''''''
        'Los destinos seleccionados
        tabla_matriz(1)(0) = punto_inicial ' Punto inicio en la primera fila
        tabla_matriz(0)(1) = punto_inicial ' Punto inicio en la primera
columna
        'Coloando los indices de los puntos en la matriz
        For a = 0 To PCriticos.RowCount - 1
            If PCriticos(1, a).Value = Enabled And PCriticos(0, a).Value <>
punto_inicial Then
                tabla_matriz.Columns.Add()
                tabla_matriz(0)(tabla_matriz.Columns.Count - 1) =
PCriticos(0, a).Value
                tabla_matriz.Rows.Add()
                tabla_matriz(tabla_matriz.Rows.Count - 1)(0) = PCriticos(0,
a).Value
            End If
        Next
        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        'Calculando la matriz de distancia euclidianas
        Dim PI As Double = 3.1415926535897931
        Dim radio As Integer
        radio = 6371000 'promedio  de radio terrestre medio, en metros
        Dim c
        Dim cx, cy, dx, dy As Double ' --> coordenadas geograficas de los
puntos criticos
        Dim dis_eucli As Integer
        tabla_matriz(1)(1) = 0
        For i = 2 To tabla_matriz.Columns.Count - 1
            cx = PTactico(2, fil_punto_inicial).Value
            cy = PTactico(3, fil_punto_inicial).Value
            For t = 0 To PCriticos.RowCount - 1
                If tabla_matriz(0)(i) = PCriticos(0, t).Value Then
                    dx = PCriticos(2, t).Value
                    dy = PCriticos(3, t).Value
                    Exit For
                End If
            Next
            dx = dx
            dy = dy
            'dis_eucli = Math.Sqrt(Math.Pow(84.8 * (cx - dx), 2) +
Math.Pow(110.56 * (cy - dy), 2)) * 1000
            c = PI / 180
            dis_eucli = 2 * PI * radio *
(Math.Asin(Math.Sqrt(Math.Pow(Math.Sin(c * (dy - cy) / 2), 2) + Math.Cos(c *
cy) * Math.Cos(c * dy) * Math.Sin(c * (dx - cx) / 2) * Math.Sin(c * (dx - cx)
/ 2))))
            tabla_matriz(1)(i) = dis_eucli
            tabla_matriz(i)(1) = dis_eucli
            dis_eucli = 0
        Next
        For n = 2 To tabla_matriz.Rows.Count - 1
            For t = 0 To PCriticos.RowCount - 1
                If tabla_matriz(n)(0) = PCriticos(0, t).Value Then
                    cx = PCriticos(2, t).Value
                    cy = PCriticos(3, t).Value
                    Exit For
                End If
```

```
            Next
            cx = cx
            cy = cy
            For m = 2 To tabla_matriz.Columns.Count - 1
                For t = 0 To PCriticos.RowCount - 1
                    If tabla_matriz(0)(m) = PCriticos(0, t).Value Then
                        dx = PCriticos(2, t).Value
                        dy = PCriticos(3, t).Value
                        Exit For
                    End If
                Next
                dx = dx
                dy = dy
                c = PI / 180
                ' dis_eucli = Math.Sqrt(Math.Pow(84.8 * (cx - dx), 2) +
Math.Pow(110.56 * (cy - dy), 2)) * 1000
                dis_eucli = 2 * PI * radio *
(Math.Asin(Math.Sqrt(Math.Pow(Math.Sin(c * (dy - cy) / 2), 2) + Math.Cos(c *
cy) * Math.Cos(c * dy) * Math.Sin(c * (dx - cx) / 2) * Math.Sin(c * (dx - cx)
/ 2))))
                tabla_matriz(n)(m) = dis_eucli
            Next
        Next
        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        ' Asignando la matriz de distancia euclidiana a los formularios de
cada algoritmo
        'Mandando al Algoritmo de Clarke y Wright
        AlgoritmoClarkeYWright.DataGridView1.DataSource = tabla_matriz
        AlgoritmoClarkeYWright.Label7.Text = NComisaria.Text
        'Mandando al Algortimo del Vecino mas Cercano
        AlgoritmoVecinoCercano.DataGridView4.DataSource = tabla_matriz
        AlgoritmoVecinoCercano.Label1.Text = NComisaria.Text
        'Mandando al Algortimo de Busqueda Tabu
        AlgoritmoBusquedaTabu.DataGridView8.DataSource = tabla_matriz
        AlgoritmoBusquedaTabu.Label1.Text = NComisaria.Text
        'Mandando al Algortimo Genetico
        AlgoritmoGenetico.DataGridView9.DataSource = tabla_matriz
        AlgoritmoGenetico.Label1.Text = NComisaria.Text
        'Mandando a MostrandoRutas
        ResultadosFinales.Label8.Text = NComisaria.Text
        MostrarRutas.Label1.Text = NComisaria.Text
        AlgoritmoClarkeYWright.Show()
    End Sub
    Private Sub BSector_Click(sender As Object, e As EventArgs) Handles
BSector.Click
        'Tabla PCriticos nuevamente vacio
        PCriticos.ColumnCount = 5
        PCriticos.RowCount = 0
        PTactico.ColumnCount = 5
        PTactico.RowCount = 0
        'Limpiando los elementos del mapa
        Mapa.Overlays.Clear()
        'Colocando el poligo completo del area de jurisdiccion
        Select Case NComisaria.Text
            Case "PIEDRA LIZA"
                Poligono_PD()
                If CheckedListBoxSectores.GetItemChecked(0) = True Then
                    'Colocando los puntos criticos en el datagrid
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC1"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC2"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC3"
```

```
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC4"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC5"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC6"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC7"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC8"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP1"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP2"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP3"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP4"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP5"
                    'Colocando Puestos Tacticos en datagrid
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT1"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT2"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT3"
                    'colocando las coordenadas geograficas de PC
                    Dim punto
                    Dim x
                    Dim y
                    For i = 0 To PCriticos.RowCount - 1
                        punto = PCriticos(0, i).Value
                        For f = 0 To
CoordenadasGeograficas.DataGridView1.RowCount - 1
                            If punto =
CoordenadasGeograficas.DataGridView1(0, f).Value Then
                                x = CoordenadasGeograficas.DataGridView1(1,
f).Value
                                y = CoordenadasGeograficas.DataGridView1(2,
f).Value
                                PCriticos(2, i).Value = x
                                PCriticos(3, i).Value = y
                                PCriticos(4, i).Value =
CoordenadasGeograficas.DataGridView1(3, f).Value
                                Exit For
                            End If
                        Next
                    Next
                    PCriticos.Columns(2).Visible = False
                    PCriticos.Columns(3).Visible = False
                    PCriticos.Columns(4).Visible = True
                    PCriticos.Columns(0).Width = 70
                    PCriticos.Columns(1).Width = 30
                    PCriticos.Columns(4).Width = 133
                    'Colocando las coordenadas geograficas de PT
                    For i = 0 To PTactico.RowCount - 1
                        punto = PTactico(0, i).Value
                        For f = 0 To
CoordenadasGeograficas.DataGridView1.RowCount - 1
                            If punto =
CoordenadasGeograficas.DataGridView1(0, f).Value Then
                                x = CoordenadasGeograficas.DataGridView1(1,
f).Value
```

```
                                y = CoordenadasGeograficas.DataGridView1(2,
f).Value
                                PTactico(2, i).Value = x
                                PTactico(3, i).Value = y
                                PTactico(4, i).Value =
CoordenadasGeograficas.DataGridView1(3, f).Value
                                Exit For
                            End If
                        Next
                    Next
                    PTactico.Columns(2).Visible = False
                    PTactico.Columns(3).Visible = False
                    PTactico.Columns(4).Visible = True
                    PTactico.Columns(0).Width = 70
                    PTactico.Columns(1).Width = 30
                    PTactico.Columns(4).Width = 133
                End If
                If CheckedListBoxSectores.GetItemChecked(1) = True Then
                    'Colocando los puntos criticos en el datagrid
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC9"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC10"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC11"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC12"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC13"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP6"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP7"
                    'Colocando Puestos Tacticos en datagrid
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT4"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT5"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT6"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT7"
                    'colocando las coordenadas geograficas de PC
                    Dim punto
                    Dim x
                    Dim y
                    For i = 0 To PCriticos.RowCount - 1
                        punto = PCriticos(0, i).Value
                        For f = 0 To
CoordenadasGeograficas.DataGridView1.RowCount - 1
                            If punto =
CoordenadasGeograficas.DataGridView1(0, f).Value Then
                                x = CoordenadasGeograficas.DataGridView1(1,
f).Value
                                y = CoordenadasGeograficas.DataGridView1(2,
f).Value
                                PCriticos(2, i).Value = x
                                PCriticos(3, i).Value = y
                                PCriticos(4, i).Value =
CoordenadasGeograficas.DataGridView1(3, f).Value
                                Exit For
                            End If
                        Next
                    Next
```

```
                    PCriticos.Columns(2).Visible = False
                    PCriticos.Columns(3).Visible = False
                    PCriticos.Columns(4).Visible = True
                    PCriticos.Columns(0).Width = 70
                    PCriticos.Columns(1).Width = 30
                    PCriticos.Columns(4).Width = 133
                    'Colocando las coordenadas geograficas de PT
                    For i = 0 To PTactico.RowCount - 1
                        punto = PTactico(0, i).Value
                        For f = 0 To
CoordenadasGeograficas.DataGridView1.RowCount - 1
                            If punto =
CoordenadasGeograficas.DataGridView1(0, f).Value Then
                                x = CoordenadasGeograficas.DataGridView1(1,
f).Value
                                y = CoordenadasGeograficas.DataGridView1(2,
f).Value
                                PTactico(2, i).Value = x
                                PTactico(3, i).Value = y
                                PTactico(4, i).Value =
CoordenadasGeograficas.DataGridView1(3, f).Value
                                Exit For
                            End If
                        Next
                    Next
                    PTactico.Columns(2).Visible = False
                    PTactico.Columns(3).Visible = False
                    PTactico.Columns(4).Visible = True
                    PTactico.Columns(0).Width = 70
                    PTactico.Columns(1).Width = 30
                    PTactico.Columns(4).Width = 133
                End If
                If CheckedListBoxSectores.GetItemChecked(2) = True Then
                    'Colocando los puntos criticos en el datagrid
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC14"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC15"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC16"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC17"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC18"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP8"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP9"
                    'Colocando Puestos Tacticos en datagrid
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT8"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT9"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT10"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT11"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT12"
                    'colocando las coordenadas geograficas de PC
                    Dim punto
                    Dim x
                    Dim y
                    For i = 0 To PCriticos.RowCount - 1
```

```
                        punto = PCriticos(0, i).Value
                        For f = 0 To
CoordenadasGeograficas.DataGridView1.RowCount - 1
                            If punto =
CoordenadasGeograficas.DataGridView1(0, f).Value Then
                                x = CoordenadasGeograficas.DataGridView1(1,
f).Value
                                y = CoordenadasGeograficas.DataGridView1(2,
f).Value
                                PCriticos(2, i).Value = x
                                PCriticos(3, i).Value = y
                                PCriticos(4, i).Value =
CoordenadasGeograficas.DataGridView1(3, f).Value
                                Exit For
                            End If
                        Next
                    Next
                    PCriticos.Columns(2).Visible = False
                    PCriticos.Columns(3).Visible = False
                    PCriticos.Columns(4).Visible = True
                    PCriticos.Columns(0).Width = 70
                    PCriticos.Columns(1).Width = 30
                    PCriticos.Columns(4).Width = 133
                    'Colocando las coordenadas geograficas de PT
                    For i = 0 To PTactico.RowCount - 1
                        punto = PTactico(0, i).Value
                        For f = 0 To
CoordenadasGeograficas.DataGridView1.RowCount - 1
                            If punto =
CoordenadasGeograficas.DataGridView1(0, f).Value Then
                                x = CoordenadasGeograficas.DataGridView1(1,
f).Value
                                y = CoordenadasGeograficas.DataGridView1(2,
f).Value
                                PTactico(2, i).Value = x
                                PTactico(3, i).Value = y
                                PTactico(4, i).Value =
CoordenadasGeograficas.DataGridView1(3, f).Value
                                Exit For
                            End If
                        Next
                    Next
                    PTactico.Columns(2).Visible = False
                    PTactico.Columns(3).Visible = False
                    PTactico.Columns(4).Visible = True
                    PTactico.Columns(0).Width = 70
                    PTactico.Columns(1).Width = 30
                    PTactico.Columns(4).Width = 133
                End If
                If CheckedListBoxSectores.GetItemChecked(3) = True Then
                    'Colocando los puntos criticos en el datagrid
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC19"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC20"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC21"
                    'Colocando Puestos Tacticos en datagrid
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT13"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT14"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT15"
```

```
                    'colocando las coordenadas geograficas de PC
                    Dim punto
                    Dim x
                    Dim y
                    For i = 0 To PCriticos.RowCount - 1
                        punto = PCriticos(0, i).Value
                        For f = 0 To
CoordenadasGeograficas.DataGridView1.RowCount - 1
                            If punto =
CoordenadasGeograficas.DataGridView1(0, f).Value Then
                                x = CoordenadasGeograficas.DataGridView1(1,
f).Value
                                y = CoordenadasGeograficas.DataGridView1(2,
f).Value
                                PCriticos(2, i).Value = x
                                PCriticos(3, i).Value = y
                                PCriticos(4, i).Value =
CoordenadasGeograficas.DataGridView1(3, f).Value
                                Exit For
                            End If
                        Next
                    Next
                    PCriticos.Columns(2).Visible = False
                    PCriticos.Columns(3).Visible = False
                    PCriticos.Columns(4).Visible = True
                    PCriticos.Columns(0).Width = 70
                    PCriticos.Columns(1).Width = 30
                    PCriticos.Columns(4).Width = 133
                    'Colocando las coordenadas geograficas de PT
                    For i = 0 To PTactico.RowCount - 1
                        punto = PTactico(0, i).Value
                        For f = 0 To
CoordenadasGeograficas.DataGridView1.RowCount - 1
                            If punto =
CoordenadasGeograficas.DataGridView1(0, f).Value Then
                                x = CoordenadasGeograficas.DataGridView1(1,
f).Value
                                y = CoordenadasGeograficas.DataGridView1(2,
f).Value
                                PTactico(2, i).Value = x
                                PTactico(3, i).Value = y
                                PTactico(4, i).Value =
CoordenadasGeograficas.DataGridView1(3, f).Value
                                Exit For
                            End If
                        Next
                    Next
                    PTactico.Columns(2).Visible = False
                    PTactico.Columns(3).Visible = False
                    PTactico.Columns(4).Visible = True
                    PTactico.Columns(0).Width = 70
                    PTactico.Columns(1).Width = 30
                    PTactico.Columns(4).Width = 133
                End If
            Case "RIMAC"
                Poligono_RI()
                If CheckedListBoxSectores.GetItemChecked(0) = True Then
                    'Colocando los puntos criticos en el datagrid
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC1"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC2"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC3"
```

```
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC4"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC5"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC6"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC7"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC8"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP1"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP2"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP3"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP4"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP5"
                    'Colocando Puestos Tacticos en datagrid
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT1"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT2"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT3"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT4"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT5"
                    'colocando las coordenadas geograficas de PC
                    Dim punto
                    Dim x
                    Dim y
                    For i = 0 To PCriticos.RowCount - 1
                        punto = PCriticos(0, i).Value
                        For f = 0 To
CoordenadasGeograficas.DataGridView2.RowCount - 1
                            If punto =
CoordenadasGeograficas.DataGridView2(0, f).Value Then
                                x = CoordenadasGeograficas.DataGridView2(1,
f).Value
                                y = CoordenadasGeograficas.DataGridView2(2,
f).Value
                                PCriticos(2, i).Value = x
                                PCriticos(3, i).Value = y
                                PCriticos(4, i).Value =
CoordenadasGeograficas.DataGridView2(3, f).Value
                                Exit For
                            End If
                        Next
                    Next
                    PCriticos.Columns(2).Visible = False
                    PCriticos.Columns(3).Visible = False
                    PCriticos.Columns(4).Visible = True
                    PCriticos.Columns(0).Width = 70
                    PCriticos.Columns(1).Width = 30
                    PCriticos.Columns(4).Width = 133
                    'Colocando las coordenadas geograficas de PR
                    For i = 0 To PTactico.RowCount - 1
                        punto = PTactico(0, i).Value
                        For f = 0 To
CoordenadasGeograficas.DataGridView2.RowCount - 1
```

```
                            If punto =
CoordenadasGeograficas.DataGridView2(0, f).Value Then
                                x = CoordenadasGeograficas.DataGridView2(1,
f).Value
                                y = CoordenadasGeograficas.DataGridView2(2,
f).Value
                                PTactico(2, i).Value = x
                                PTactico(3, i).Value = y
                                PTactico(4, i).Value =
CoordenadasGeograficas.DataGridView2(3, f).Value
                                Exit For
                            End If
                        Next
                    Next
                    PTactico.Columns(2).Visible = False
                    PTactico.Columns(3).Visible = False
                    PTactico.Columns(4).Visible = True
                    PTactico.Columns(0).Width = 70
                    PTactico.Columns(1).Width = 30
                    PTactico.Columns(4).Width = 133
                End If
                If CheckedListBoxSectores.GetItemChecked(1) = True Then
                    'Colocando los puntos criticos en el datagrid
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC9"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC10"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC11"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC12"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC13"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC14"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP6"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP7"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP8"
                    'Colocando Puestos Tacticos en datagrid
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT6"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT7"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT8"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT9"
                    'colocando las coordenadas geograficas de PC
                    Dim punto
                    Dim x
                    Dim y
                    For i = 0 To PCriticos.RowCount - 1
                        punto = PCriticos(0, i).Value
                        For f = 0 To
CoordenadasGeograficas.DataGridView2.RowCount - 1
                            If punto =
CoordenadasGeograficas.DataGridView2(0, f).Value Then
                                x = CoordenadasGeograficas.DataGridView2(1,
f).Value
                                y = CoordenadasGeograficas.DataGridView2(2,
f).Value
```

```
                                PCriticos(2, i).Value = x
                                PCriticos(3, i).Value = y
                                PCriticos(4, i).Value =
CoordenadasGeograficas.DataGridView2(3, f).Value
                                Exit For
                            End If
                        Next
                    Next
                    PCriticos.Columns(2).Visible = False
                    PCriticos.Columns(3).Visible = False
                    PCriticos.Columns(4).Visible = True
                    PCriticos.Columns(0).Width = 70
                    PCriticos.Columns(1).Width = 30
                    PCriticos.Columns(4).Width = 133
                    'Colocando las coordenadas geograficas
                    For i = 0 To PTactico.RowCount - 1
                        punto = PTactico(0, i).Value
                        For f = 0 To
CoordenadasGeograficas.DataGridView2.RowCount - 1
                            If punto =
CoordenadasGeograficas.DataGridView2(0, f).Value Then
                                x = CoordenadasGeograficas.DataGridView2(1,
f).Value
                                y = CoordenadasGeograficas.DataGridView2(2,
f).Value
                                PTactico(2, i).Value = x
                                PTactico(3, i).Value = y
                                PTactico(4, i).Value =
CoordenadasGeograficas.DataGridView2(3, f).Value
                                Exit For
                            End If
                        Next
                    Next
                    PTactico.Columns(2).Visible = False
                    PTactico.Columns(3).Visible = False
                    PTactico.Columns(4).Visible = True
                    PTactico.Columns(0).Width = 70
                    PTactico.Columns(1).Width = 30
                    PTactico.Columns(4).Width = 133
                End If
                If CheckedListBoxSectores.GetItemChecked(2) = True Then
                    'Colocando los puntos criticos en el datagrid
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC15"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC16"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC17"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC18"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC19"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC20"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC21"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC22"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC23"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC24"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC25"
```

```
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC26"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC27"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP9"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP10"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP11"
                    'Colocando Puestos Tacticos en datagrid
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT10"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT11"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT12"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT13"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT14"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT15"
                    'colocando las coordenadas geograficas de PC
                    Dim punto
                    Dim x
                    Dim y
                    For i = 0 To PCriticos.RowCount - 1
                        punto = PCriticos(0, i).Value
                        For f = 0 To
CoordenadasGeograficas.DataGridView2.RowCount - 1
                            If punto =
CoordenadasGeograficas.DataGridView2(0, f).Value Then
                                x = CoordenadasGeograficas.DataGridView2(1,
f).Value
                                y = CoordenadasGeograficas.DataGridView2(2,
f).Value
                                PCriticos(2, i).Value = x
                                PCriticos(3, i).Value = y
                                PCriticos(4, i).Value =
CoordenadasGeograficas.DataGridView2(3, f).Value
                                Exit For
                            End If
                        Next
                    Next
                    PCriticos.Columns(2).Visible = False
                    PCriticos.Columns(3).Visible = False
                    PCriticos.Columns(4).Visible = True
                    PCriticos.Columns(0).Width = 70
                    PCriticos.Columns(1).Width = 30
                    PCriticos.Columns(4).Width = 133
                    'Colocando las coordenadas geograficas
                    For i = 0 To PTactico.RowCount - 1
                        punto = PTactico(0, i).Value
                        For f = 0 To
CoordenadasGeograficas.DataGridView2.RowCount - 1
                            If punto =
CoordenadasGeograficas.DataGridView2(0, f).Value Then
                                x = CoordenadasGeograficas.DataGridView2(1,
f).Value
                                y = CoordenadasGeograficas.DataGridView2(2,
f).Value
                                PTactico(2, i).Value = x
                                PTactico(3, i).Value = y
```

```
                                PTactico(4, i).Value =
CoordenadasGeograficas.DataGridView2(3, f).Value
                                Exit For
                            End If
                        Next
                    Next
                    PTactico.Columns(2).Visible = False
                    PTactico.Columns(3).Visible = False
                    PTactico.Columns(4).Visible = True
                    PTactico.Columns(0).Width = 70
                    PTactico.Columns(1).Width = 30
                    PTactico.Columns(4).Width = 133
                End If
                If CheckedListBoxSectores.GetItemChecked(3) = True Then
                    'Colocando los puntos criticos en el datagrid
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC28"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC29"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC30"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC31"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC32"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC33"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC34"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP12"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP13"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP14"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP15"
                    'Colocando Puestos Tacticos en datagrid
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT16"
                    PTactico(0, PTactico.RowCount - 1).Value = "PT17"
                    PTactico(0, PTactico.RowCount - 1).Value = "PT18"
                    PTactico(0, PTactico.RowCount - 1).Value = "PT19"
                    'colocando las coordenadas geograficas de PC
                    Dim punto
                    Dim x
                    Dim y
                    For i = 0 To PCriticos.RowCount - 1
                        punto = PCriticos(0, i).Value
                        For f = 0 To
CoordenadasGeograficas.DataGridView2.RowCount - 1
                            If punto =
CoordenadasGeograficas.DataGridView2(0, f).Value Then
                                x = CoordenadasGeograficas.DataGridView2(1,
f).Value
                                y = CoordenadasGeograficas.DataGridView2(2,
f).Value
                                PCriticos(2, i).Value = x
                                PCriticos(3, i).Value = y
                                PCriticos(4, i).Value =
CoordenadasGeograficas.DataGridView2(3, f).Value
                                Exit For
                            End If
                        Next
```

```
                    Next
                    PCriticos.Columns(2).Visible = False
                    PCriticos.Columns(3).Visible = False
                    PCriticos.Columns(4).Visible = True
                    PCriticos.Columns(0).Width = 70
                    PCriticos.Columns(1).Width = 30
                    PCriticos.Columns(4).Width = 133
                    'Colocando las coordenadas geograficas de PT
                    For i = 0 To PTactico.RowCount - 1
                        punto = PTactico(0, i).Value
                        For f = 0 To
CoordenadasGeograficas.DataGridView2.RowCount - 1
                            If punto =
CoordenadasGeograficas.DataGridView2(0, f).Value Then
                                x = CoordenadasGeograficas.DataGridView2(1,
f).Value
                                y = CoordenadasGeograficas.DataGridView2(2,
f).Value
                                PTactico(2, i).Value = x
                                PTactico(3, i).Value = y
                                PTactico(4, i).Value =
CoordenadasGeograficas.DataGridView2(3, f).Value
                                Exit For
                            End If
                        Next
                    Next
                    PTactico.Columns(2).Visible = False
                    PTactico.Columns(3).Visible = False
                    PTactico.Columns(4).Visible = True
                    PTactico.Columns(0).Width = 70
                    PTactico.Columns(1).Width = 30
                    PTactico.Columns(4).Width = 133
                End If
                If CheckedListBoxSectores.GetItemChecked(4) = True Then
                    'Colocando los puntos criticos en el datagrid
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC35"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC36"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC37"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC38"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC39"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP16"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP17"
                    'Colocando Puestos Tacticos en datagrid
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT20"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT21"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT22"
                    'colocando las coordenadas geograficas
                    Dim punto
                    Dim x
                    Dim y
                    For i = 0 To PCriticos.RowCount - 1
                        punto = PCriticos(0, i).Value
                        For f = 0 To
CoordenadasGeograficas.DataGridView2.RowCount - 1
```

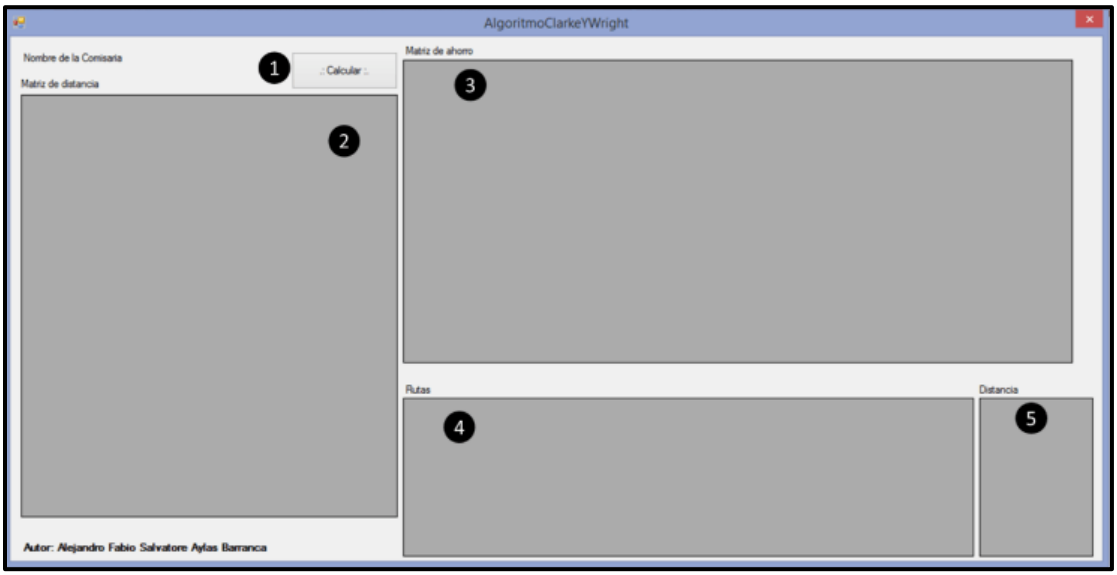
```
                            If punto =
CoordenadasGeograficas.DataGridView2(0, f).Value Then
                                x = CoordenadasGeograficas.DataGridView2(1,
f).Value
                                y = CoordenadasGeograficas.DataGridView2(2,
f).Value
                                PCriticos(2, i).Value = x
                                PCriticos(3, i).Value = y
                                PCriticos(4, i).Value =
CoordenadasGeograficas.DataGridView2(3, f).Value
                                Exit For
                            End If
                        Next
                    Next
                    PCriticos.Columns(2).Visible = False
                    PCriticos.Columns(3).Visible = False
                    PCriticos.Columns(4).Visible = True
                    PCriticos.Columns(0).Width = 70
                    PCriticos.Columns(1).Width = 30
                    PCriticos.Columns(4).Width = 133
                    'Colocando las coordenadas geograficas de PT
                    For i = 0 To PTactico.RowCount - 1
                        punto = PTactico(0, i).Value
                        For f = 0 To
CoordenadasGeograficas.DataGridView2.RowCount - 1
                            If punto =
CoordenadasGeograficas.DataGridView2(0, f).Value Then
                                x = CoordenadasGeograficas.DataGridView2(1,
f).Value
                                y = CoordenadasGeograficas.DataGridView2(2,
f).Value
                                PTactico(2, i).Value = x
                                PTactico(3, i).Value = y
                                PTactico(4, i).Value =
CoordenadasGeograficas.DataGridView2(3, f).Value
                                Exit For
                            End If
                        Next
                    Next
                    PTactico.Columns(2).Visible = False
                    PTactico.Columns(3).Visible = False
                    PTactico.Columns(4).Visible = True
                    PTactico.Columns(0).Width = 70
                    PTactico.Columns(1).Width = 30
                    PTactico.Columns(4).Width = 133
                End If
                If CheckedListBoxSectores.GetItemChecked(5) = True Then
                    'Colocando los puntos criticos en el datagrid
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP18"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP19"
                    'Colocando Puestos Tacticos en datagrid
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT23"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT24"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT25"
                    'colocando las coordenadas geograficas
                    Dim punto
                    Dim x
                    Dim y
                    For i = 0 To PCriticos.RowCount - 1
```

```
                        punto = PCriticos(0, i).Value
                        For f = 0 To
CoordenadasGeograficas.DataGridView2.RowCount - 1
                            If punto =
CoordenadasGeograficas.DataGridView2(0, f).Value Then
                                x = CoordenadasGeograficas.DataGridView2(1,
f).Value
                                y = CoordenadasGeograficas.DataGridView2(2,
f).Value
                                PCriticos(2, i).Value = x
                                PCriticos(3, i).Value = y
                                PCriticos(4, i).Value =
CoordenadasGeograficas.DataGridView2(3, f).Value
                                Exit For
                            End If
                        Next
                    Next
                    PCriticos.Columns(2).Visible = False
                    PCriticos.Columns(3).Visible = False
                    PCriticos.Columns(4).Visible = True
                    PCriticos.Columns(0).Width = 70
                    PCriticos.Columns(1).Width = 30
                    PCriticos.Columns(4).Width = 133
                    'Colocando las coordenadas geograficas de PT
                    For i = 0 To PTactico.RowCount - 1
                        punto = PTactico(0, i).Value
                        For f = 0 To
CoordenadasGeograficas.DataGridView2.RowCount - 1
                            If punto =
CoordenadasGeograficas.DataGridView2(0, f).Value Then
                                x = CoordenadasGeograficas.DataGridView2(1,
f).Value
                                y = CoordenadasGeograficas.DataGridView2(2,
f).Value
                                PTactico(2, i).Value = x
                                PTactico(3, i).Value = y
                                PTactico(4, i).Value =
CoordenadasGeograficas.DataGridView2(3, f).Value
                                Exit For
                            End If
                        Next
                    Next
                    PTactico.Columns(2).Visible = False
                    PTactico.Columns(3).Visible = False
                    PTactico.Columns(4).Visible = True
                    PTactico.Columns(0).Width = 70
                    PTactico.Columns(1).Width = 30
                    PTactico.Columns(4).Width = 133
                End If
            Case "MANZANO"
                Poligono_MA()
                If CheckedListBoxSectores.GetItemChecked(0) = True Then
                    'Colocando los puntos criticos en el datagrid
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC1"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC2"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC3"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC4"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP1"
                    'Colocando Puestos Tacticos en datagrid
```

```
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT1"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT2"
                    'colocando las coordenadas geograficas de PC
                    Dim punto
                    Dim x
                    Dim y
                    For i = 0 To PCriticos.RowCount - 1
                        punto = PCriticos(0, i).Value
                        For f = 0 To
CoordenadasGeograficas.DataGridView3.RowCount - 1
                            If punto =
CoordenadasGeograficas.DataGridView3(0, f).Value Then
                                x = CoordenadasGeograficas.DataGridView3(1,
f).Value
                                y = CoordenadasGeograficas.DataGridView3(2,
f).Value
                                PCriticos(2, i).Value = x
                                PCriticos(3, i).Value = y
                                PCriticos(4, i).Value =
CoordenadasGeograficas.DataGridView3(3, f).Value
                                Exit For
                            End If
                        Next
                    Next
                    PCriticos.Columns(2).Visible = False
                    PCriticos.Columns(3).Visible = False
                    PCriticos.Columns(4).Visible = True
                    PCriticos.Columns(0).Width = 70
                    PCriticos.Columns(1).Width = 30
                    PCriticos.Columns(4).Width = 133
                    'Colocando las coordenadas geograficas de PT
                    For i = 0 To PTactico.RowCount - 1
                        punto = PTactico(0, i).Value
                        For f = 0 To
CoordenadasGeograficas.DataGridView3.RowCount - 1
                            If punto =
CoordenadasGeograficas.DataGridView3(0, f).Value Then
                                x = CoordenadasGeograficas.DataGridView3(1,
f).Value
                                y = CoordenadasGeograficas.DataGridView3(2,
f).Value
                                PTactico(2, i).Value = x
                                PTactico(3, i).Value = y
                                PTactico(4, i).Value =
CoordenadasGeograficas.DataGridView3(3, f).Value
                                Exit For
                            End If
                        Next
                    Next
                    PTactico.Columns(2).Visible = False
                    PTactico.Columns(3).Visible = False
                    PTactico.Columns(4).Visible = True
                    PTactico.Columns(0).Width = 70
                    PTactico.Columns(1).Width = 30
                    PTactico.Columns(4).Width = 133
                End If
                If CheckedListBoxSectores.GetItemChecked(1) = True Then
                    'Colocando los puntos criticos en el datagrid
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC5"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC6"
```

```
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC7"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC8"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC9"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC10"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC11"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PC12"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP2"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP3"
                    PCriticos.Rows.Add()
                    PCriticos(0, PCriticos.RowCount - 1).Value = "PP4"
                    'Colocando Puestos Tacticos en datagrid
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT3"
                    PTactico.Rows.Add()
                    PTactico(0, PTactico.RowCount - 1).Value = "PT4"
                    'colocando las coordenadas geograficas de PC
                    Dim punto
                    Dim x
                    Dim y
                    For i = 0 To PCriticos.RowCount - 1
                        punto = PCriticos(0, i).Value
                        For f = 0 To
CoordenadasGeograficas.DataGridView3.RowCount - 1
                            If punto =
CoordenadasGeograficas.DataGridView3(0, f).Value Then
                            x = CoordenadasGeograficas.DataGridView3(1,
f).Value
                            y = CoordenadasGeograficas.DataGridView3(2,
f).Value
                            PCriticos(2, i).Value = x
                            PCriticos(3, i).Value = y
                            PCriticos(4, i).Value =
CoordenadasGeograficas.DataGridView3(3, f).Value
                            Exit For
                        End If
                    End Sub
                End Class
```

Anexo 10.4. Formulario AlgoritmoClarkeYWright



Este formulario es llamado *AlgoritmoClarkeYWright* los controles presentes son: en (1) está *Button1*, en (2) un *DataGridView1,* en (3) un *DataGridView2*, en (4) un *DataGridView3* y en (5) está *DataGridView4.* Los códigos de programación son:

```
Public Class AlgoritmoClarkeYWright
    Dim capacidad As Integer = Comisaria.NumPuntos.Text '---> numero maximo
de destinos para visitar por ruta
    Private Sub AlgoritmoClarkeYWright_Load(sender As Object, e As EventArgs)
Handles MyBase.Load
        Button1.PerformClick()
    End Sub
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        'Construllendo la matriz de ahorro
        DataGridView2.ColumnCount = DataGridView1.ColumnCount
        DataGridView2.RowCount = DataGridView1.RowCount
        Dim punto_inicial = DataGridView1(0, 1).Value
        'Punto inicial
        DataGridView2(0, DataGridView2.RowCount - 1).Value = punto_inicial
        DataGridView2(DataGridView2.ColumnCount - 1, 0).Value = punto_inicial
        'Colocando los nombres de los demas puntos
        For i = 2 To DataGridView1.RowCount - 1
            'fila
            DataGridView2(0, i - 1).Value = DataGridView1(0, i).Value
            'columna
            DataGridView2(i - 1, 0).Value = DataGridView1(i, 0).Value
        Next
        'Hallando el ahorro
        For i = 1 To DataGridView2.RowCount - 1
            For j = 1 To DataGridView2.ColumnCount - 1
                DataGridView2(j, i).Value = 0
            Next
        Next
        Dim ahorro As Integer
        Dim d1, d2
        Dim fil_d1, colum_d2 As Integer
        Dim di0, dj0, dij As Integer
        For i = 1 To DataGridView2.RowCount - 3
```

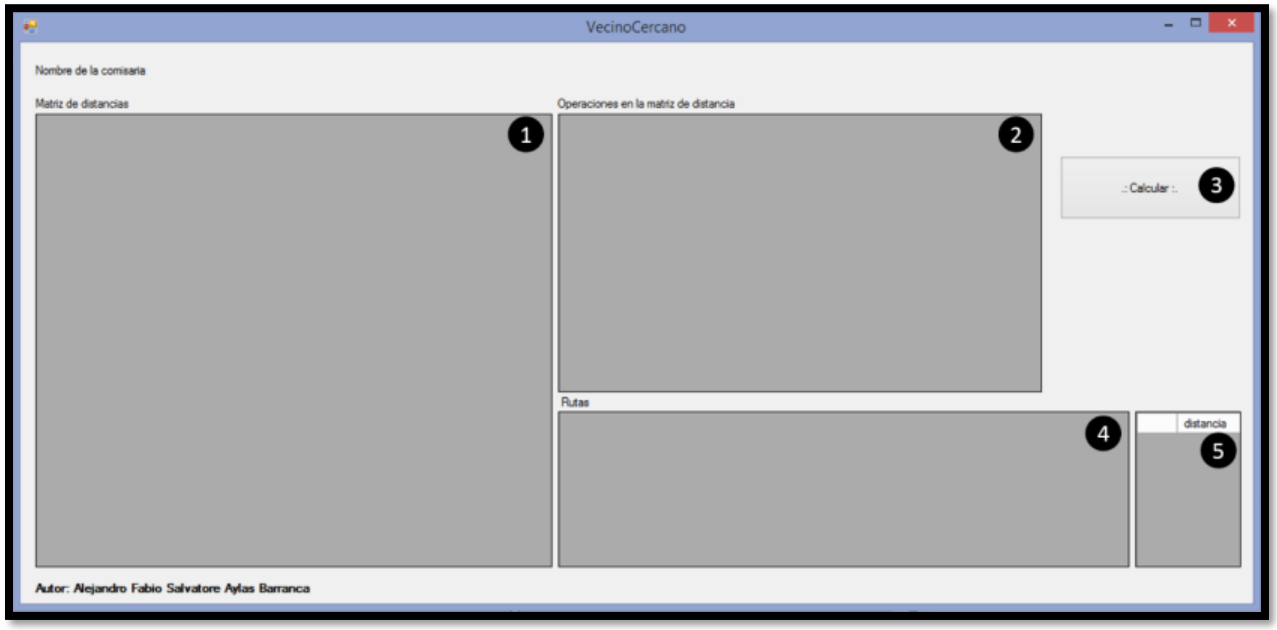
```
            For j = i + 1 To DataGridView2.ColumnCount - 2
                d1 = DataGridView2(0, i).Value
                d2 = DataGridView2(j, 0).Value
                'Buscando D1 en la matriz de distancias
                For f = 1 To DataGridView1.RowCount - 1
                    If d1 = DataGridView1(0, f).Value Then
                        fil_d1 = f
                        Exit For
                    End If
                Next
                fil_d1 = fil_d1
                di0 = DataGridView1(1, fil_d1).Value
                'Buscando D2 en la matriz de distancias
                For f = 1 To DataGridView1.ColumnCount - 1
                    If d2 = DataGridView1(f, 0).Value Then
                        colum_d2 = f
                        Exit For
                    End If
                Next
                colum_d2 = colum_d2
                dj0 = DataGridView1(colum_d2, 1).Value
                dij = DataGridView1(colum_d2, fil_d1).Value
                ahorro = di0 + dj0 - dij
                DataGridView2(j, i).Value = ahorro
                DataGridView2(i, j).Value = ahorro
            Next
        Next
        '''''''''''''''''''''''''''''''''
        'Armando el numero de columnas de la tabla de rutas
        DataGridView3.ColumnCount = capacidad + 2
        'Maximo ahorro general
        Dim max_ahorro As Integer
        Dim fil_max_ahorro As Integer
        Dim colum_max_ahorro As Integer
        Dim k = 1
        While k = 1
            'Buscando el ahorro mayor para crear una nueva ruta
            max_ahorro = -1
            For i = 1 To DataGridView2.RowCount - 2
                For j = 1 To DataGridView2.ColumnCount - 2
                    If max_ahorro <= DataGridView2(j, i).Value Then
                        max_ahorro = DataGridView2(j, i).Value
                        fil_max_ahorro = i
                        colum_max_ahorro = j
                    End If
                Next
            Next
            max_ahorro = max_ahorro
            fil_max_ahorro = fil_max_ahorro
            colum_max_ahorro = colum_max_ahorro
            If max_ahorro <> -1 Then
                'Colocando una nueva fila para una nueva ruta
                DataGridView3.Rows.Add()
                For i = 0 To DataGridView3.ColumnCount - 1
                    DataGridView3(i, DataGridView3.RowCount - 1).Value = 0
                Next
                DataGridView3(0, DataGridView3.RowCount - 1).Value =
punto_inicial
                DataGridView3(1, DataGridView3.RowCount - 1).Value =
DataGridView2(0, fil_max_ahorro).Value
                DataGridView3(2, DataGridView3.RowCount - 1).Value =
DataGridView2(colum_max_ahorro, 0).Value
                'Haciendo de -1 toda la columna de ese destino que se añadio
                For j = 1 To DataGridView2.RowCount - 2
```

```
                    DataGridView2(fil_max_ahorro, j).Value = -1
                Next
                'Haciendo de -1 toda la fila de ese destino que se añadio
                For j = 1 To DataGridView2.ColumnCount - 2
                    DataGridView2(j, fil_max_ahorro).Value = -1
                Next
                'Añadiendo los demas destinos
                For i = 2 To DataGridView3.ColumnCount - 3
                    'Punto actual de la ruta
                    Dim punto = DataGridView3(i, DataGridView3.RowCount -
1).Value
                    'Buscar en que fila de la matriz de ahorro se encuentra
el Punto Actual
                    Dim fil_destino As Integer
                    For j = 1 To DataGridView2.RowCount - 1
                        If punto = DataGridView2(0, j).Value Then
                            fil_destino = j
                            Exit For
                        End If
                    Next
                    fil_destino = fil_destino
                    'Buscar el mayor de esa fila
                    Dim maxi As Integer = -1
                    Dim colu_maxi As Integer
                    For j = 1 To DataGridView2.ColumnCount - 2
                        If DataGridView2(j, fil_destino).Value > maxi And j
<> fil_destino Then
                            colu_maxi = j
                            maxi = DataGridView2(j, fil_destino).Value
                        End If
                    Next
                    maxi = maxi
                    colu_maxi = colu_maxi
                    If maxi <> -1 Then
                        'Asigando a la ruta el destino encontrado
                        DataGridView3(i + 1, DataGridView3.RowCount -
1).Value = DataGridView2(colu_maxi, 0).Value
                        'Haciendo de -1 toda la columna del destino anterior
                        For j = 1 To DataGridView2.RowCount - 2
                            DataGridView2(fil_destino, j).Value = -1
                        Next
                        'Haciendo de -1 toda la fila del destino anterior
                        For j = 1 To DataGridView2.ColumnCount - 2
                            DataGridView2(j, fil_destino).Value = -1
                        Next
                    Else
                        Exit For
                    End If
                Next
                'Tachando la fila y columna del ultimo punto añadido a la
ruta
                Dim ultimo_punto
                For i = 0 To DataGridView3.ColumnCount - 1
                    If DataGridView3(i, DataGridView3.RowCount - 1).Value =
"0".ToString Then
                        ultimo_punto = DataGridView3(i - 1,
DataGridView3.RowCount - 1).Value
                        Exit For
                    End If
                Next
                ultimo_punto = ultimo_punto
                'Buscando ultimo punto en la matriz de ahorro
                For i = 1 To DataGridView2.ColumnCount - 2
                    If DataGridView2(i, 0).Value = ultimo_punto Then
```

```
                        'Haciendo de -1 toda la columna del destino anterior
                        For j = 1 To DataGridView2.RowCount - 2
                            DataGridView2(i, j).Value = -1
                        Next
                        'Haciendo de -1 toda la fila del destino anterior
                        For j = 1 To DataGridView2.ColumnCount - 2
                            DataGridView2(j, i).Value = -1
                        Next
                        Exit For
                    End If
                Next
            Else
                Exit While
            End If
        End While
        'Buscar si se repiten dos puntos en la ultima fila
        For i = 0 To DataGridView3.ColumnCount - 1
            If DataGridView3(i, DataGridView3.RowCount - 1).Value =
"0".ToString Then
                If DataGridView3(i - 1, DataGridView3.RowCount - 1).Value =
DataGridView3(i - 2, DataGridView3.RowCount - 1).Value Then
                    DataGridView3(i - 1, DataGridView3.RowCount - 1).Value =
0
                End If
                Exit For
            End If
        Next
        'Cortando la ultima fila inecesaria de las rutas ---> si es que
hubiera
        If DataGridView3(1, DataGridView3.RowCount - 1).Value = 0.ToString
Then
            DataGridView3.RowCount = DataGridView3.RowCount - 1
        End If
        'Colocando el punto de partida en el destino final de cada ruta
        For i = 0 To DataGridView3.RowCount - 1
            For j = 0 To DataGridView3.ColumnCount - 1
                If DataGridView3(j, i).Value = 0.ToString Then
                    DataGridView3(j, i).Value = punto_inicial
                    Exit For
                End If
            Next
        Next
        'Calculando las distancias para cada ruta
        Dim dista_acumulada As Integer = 0
        For i = 0 To DataGridView3.RowCount - 1
            DataGridView4.Rows.Add()
            'Buscar hasta que columna se va a comparar
            Dim indi_column As Integer = DataGridView3.ColumnCount - 1
            For a = 0 To DataGridView3.ColumnCount - 1
                If DataGridView3(a, i).Value = "0" Then
                    indi_column = a - 1
                    Exit For
                End If
            Next
            indi_column = indi_column
            For j = 0 To indi_column - 1
                d1 = DataGridView3(j, i).Value
                d2 = DataGridView3(j + 1, i).Value
                'Buscar d1 en la fila de matriz distancia
                For v = 1 To DataGridView1.RowCount - 1
                    If d1 = DataGridView1(0, v).Value Then
                        fil_d1 = v
                        Exit For
                    End If
```

```
                Next
                fil_d1 = fil_d1
                'Buscar d2 en la columna de matriz distancia
                For v = 1 To DataGridView1.ColumnCount - 1
                    If d2 = DataGridView1(v, 0).Value Then
                        colum_d2 = v
                        Exit For
                    End If
                Next
                colum_d2 = colum_d2
                dista_acumulada = dista_acumulada + DataGridView1(colum_d2,
fil_d1).Value
            Next
            DataGridView4(0, DataGridView4.RowCount - 1).Value =
dista_acumulada
            dista_acumulada = 0
        Next
        '-------------------------------------------------------------------
------------
        'Mandando la solucion del ahorro al formulario de Resultados Finales

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
''''
        For i = 0 To DataGridView3.RowCount - 1 'recorrido de soluciones
            'Agregando fila al datagrid de ahorro de formulario Resultados
Finales
            ResultadosFinales.DataGridView1.Rows.Add()
            ResultadosFinales.DataGridView2.Rows.Add()
            'Pegando la ruta en la data Resultados Finales
            For p = 0 To DataGridView3.ColumnCount - 1

ResultadosFinales.DataGridView1.Rows(ResultadosFinales.DataGridView1.RowCount
- 1).Cells(p).Value = DataGridView3.Rows(i).Cells(p).Value
            Next
            'Pegando la distancia en la data Resultados Finales

ResultadosFinales.DataGridView2.Rows(ResultadosFinales.DataGridView2.RowCount
- 1).Cells(0).Value = DataGridView4.Rows(i).Cells(0).Value
        Next
        AlgoritmoVecinoCercano.Show()
    End Sub
End Class
```

Anexo 10.5. Formulario AlgoritmoVecinoCercano



Este formulario es llamado *AlgoritmoVecinoCercano* los controles presentes son: en (1) está un DataGridView llamado *DataGridView4*, en (2) un DataGridView llamado *DataGridView1,* en (3) está un Button llamado *Button1,* en (4) se encuentra un DataGridView llamado *DataGridView2* y en (5) está un DataGridView llamado *DataGridView3.* Los códigos de programación son:

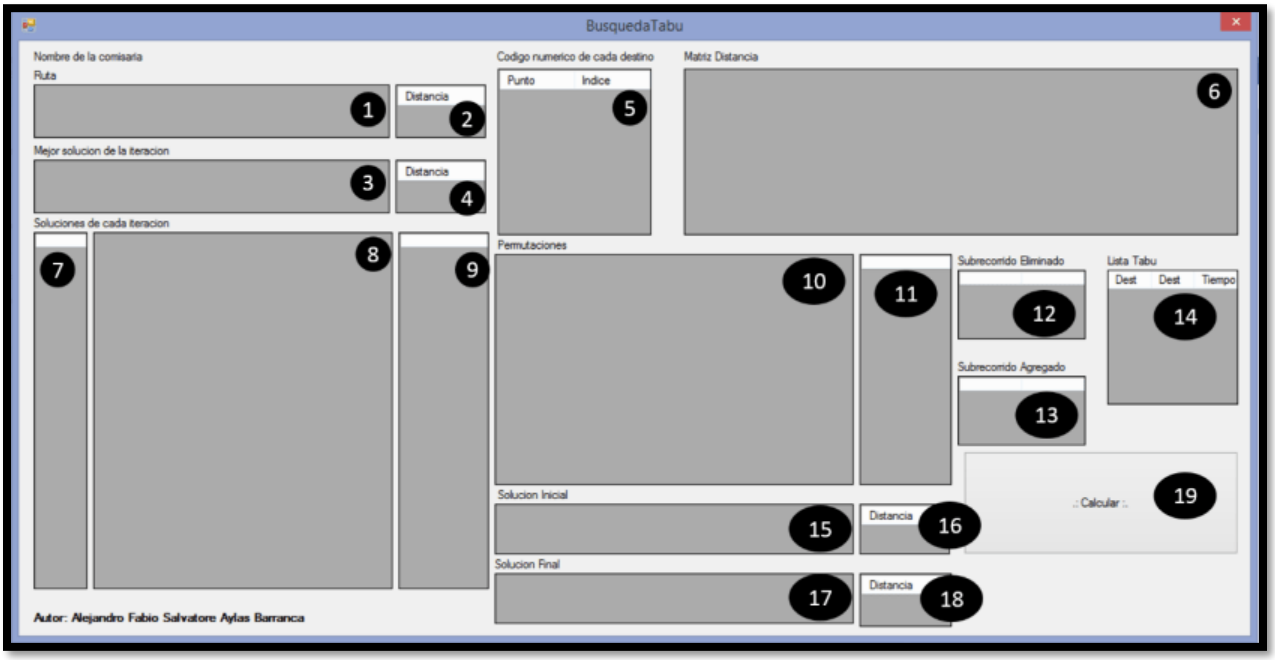
```
Public Class AlgoritmoVecinoCercano
    Dim capacidad As Integer = Comisaria.NumPuntos.Text '---> numero maximo
de destinos para visitar por ruta
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        AlgoritmoClarkeYWright.Close()
        '''''''''''''''''''''''''''''''''
        'Armando la matriz de distancia donde se hara las operaciones
        DataGridView1.ColumnCount = DataGridView4.ColumnCount
        DataGridView1.RowCount = DataGridView4.RowCount
        For i = 0 To DataGridView4.ColumnCount - 1
            For j = 0 To DataGridView4.RowCount - 1
                DataGridView1(j, i).Value = DataGridView4(j, i).Value
            Next
        Next
        '''''''''''''''''''''''''''''''''
        'Aplicando Algoritmo Vecino mas Cercano
        '''''''''''''''''''''''''''''''''
        Dim Punto_inicial = DataGridView1(0, 1).Value
        Dim fil_Po As Integer = 1 'ubicacion de la fila de la matriz donde se
encuentra el punto de partida
        '''''''''''''''''''''''''''''''''
        'Armando el numero de columnas de la tabla de rutas
        DataGridView2.ColumnCount = capacidad + 2
        'Haciendo la columna del punto de partida de ceros para que no se
repita en la ruta
        For j = 1 To DataGridView1.RowCount - 1
```

```
            DataGridView1(fil_Po, j).Value = 0
        Next
        '''''''''''''''''''''''''''''''
        'Hallando las rutas
        '''''''''''''''''''''''''''''''
        Dim k = 1
        While k = 1
            Dim salida = 0
            'Colocando una nueva fila para una nueva ruta
            DataGridView2.Rows.Add()
            For i = 0 To DataGridView2.ColumnCount - 1
                DataGridView2(i, DataGridView2.RowCount - 1).Value = 0
            Next
            DataGridView2(0, DataGridView2.RowCount - 1).Value =
Punto_inicial
            'Ejecucion del algoritmo
            For i = 0 To DataGridView2.ColumnCount - 3
                Dim punto = DataGridView2(i, DataGridView2.RowCount -
1).Value 'Punto actual de la ruta
                'Buscar en que fila de la matriz de distancia se encuentra el
Punto Actual
                Dim fil_destino As Integer
                For j = 1 To DataGridView1.RowCount - 1
                    If punto = DataGridView1(0, j).Value Then
                        fil_destino = j
                        Exit For
                    End If
                Next
                fil_destino = fil_destino
                'Buscar el menor de esa fila
                Dim mini As Integer = 9999999
                Dim colu_mini As Integer
                For j = 1 To DataGridView1.ColumnCount - 1
                    If DataGridView1(j, fil_destino).Value < mini And
DataGridView1(j, fil_destino).Value <> 0 Then
                        colu_mini = j
                        mini = DataGridView1(j, fil_destino).Value
                    End If
                Next
                colu_mini = colu_mini
                If mini <> 9999999 Then
                    'Asigando a la ruta el destino encontrado
                    DataGridView2(i + 1, DataGridView2.RowCount - 1).Value =
DataGridView1(colu_mini, 0).Value
                    'Haciendo de ceros toda la columna de ese destino que se
añadio
                    For j = 1 To DataGridView1.RowCount - 1
                        DataGridView1(colu_mini, j).Value = 0
                    Next
                Else
                    salida = 1
                    Exit For
                End If
            Next
            If salida = 1 Then
                Exit While
            End If
        End While
        'Cortando la ultima fila inecesaria de las rutas ---> si es que
hubiera
        If DataGridView2(1, DataGridView2.RowCount - 1).Value = 0.ToString
Then
            DataGridView2.RowCount = DataGridView2.RowCount - 1
        End If
```

```
        'Colocando el punto de partida en el destino final de cada ruta
        For i = 0 To DataGridView2.RowCount - 1
            For j = 0 To DataGridView2.ColumnCount - 1
                If DataGridView2(j, i).Value = 0.ToString Then
                    DataGridView2(j, i).Value = Punto_inicial
                    Exit For
                End If
            Next
        Next
        'Calculando las distancia de cada ruta
        For p = 0 To DataGridView2.RowCount - 1 ' recorrido por filas de la
tabla de rutas
            DataGridView3.Rows.Add()
            Dim colum As Integer = 0
            'hasta que columna sera el recorrido
            For j = 0 To DataGridView2.ColumnCount - 1
                If DataGridView2(j, p).Value = 0.ToString Then
                    colum = j
                    Exit For
                End If
            Next
            colum = colum
            If colum <> 0 Then 'aparecio un cero en la fila
                Dim distancias As Integer = 0
                For f = 0 To colum - 2
                    'buscando la fila del punto actual en la matriz de
distancia
                    Dim posicion_f As Integer
                    For i = 1 To DataGridView4.RowCount - 1
                        If DataGridView4(0, i).Value = DataGridView2(f,
p).Value Then
                            posicion_f = i
                            Exit For
                        End If
                    Next
                    posicion_f = posicion_f
                    'buscando la columna del punto siguiente al actual en la
matriz de distancia
                    Dim posicion_f1 As Integer
                    For i = 1 To DataGridView4.ColumnCount - 1
                        If DataGridView4(i, 0).Value = DataGridView2(f + 1,
p).Value Then
                            posicion_f1 = i
                            Exit For
                        End If
                    Next
                    posicion_f1 = posicion_f1
                    distancias = distancias + DataGridView4(posicion_f1,
posicion_f).Value
                Next
                distancias = distancias
                DataGridView3(0, DataGridView3.RowCount - 1).Value =
distancias
            ElseIf colum = 0 Then ' no hay ningun cero en la fila
                Dim distancias As Integer = 0
                For f = 0 To DataGridView2.ColumnCount - 2
                    'buscando la fila del punto actual en la matriz de
distancia
                    Dim posicion_f As Integer
                    For i = 1 To DataGridView4.RowCount - 1
                        If DataGridView4(0, i).Value = DataGridView2(f,
p).Value Then
                            posicion_f = i
                            Exit For
```

```
                    End If
                Next
                posicion_f = posicion_f
                'buscando la columna del punto siguiente al actual en la
matriz de distancia
                Dim posicion_f1 As Integer
                For i = 1 To DataGridView4.ColumnCount - 1
                    If DataGridView4(i, 0).Value = DataGridView2(f + 1,
p).Value Then
                        posicion_f1 = i
                        Exit For
                    End If
                Next
                posicion_f1 = posicion_f1
                distancias = distancias + DataGridView4(posicion_f1,
posicion_f).Value
            Next
            distancias = distancias
            DataGridView3(0, DataGridView3.RowCount - 1).Value =
distancias
        End If
    Next
    '''''''''''''''''''''''''''''''''''''''''''''''''''
    'Mandando las soluciones a formulario de Resultados finales
    '''''''''''''''''''''''''''''''''''''''''''''''''''
    For i = 0 To DataGridView2.RowCount - 1 'recorrido de soluciones
        'Agregando fila al datagrid de Vecino mas Cercano de formulario
Resultados Finales
        ResultadosFinales.DataGridView7.Rows.Add()
        ResultadosFinales.DataGridView8.Rows.Add()
        'Pegando la ruta en la data Resultados Finales
        For p = 0 To DataGridView2.ColumnCount - 1

ResultadosFinales.DataGridView7.Rows(ResultadosFinales.DataGridView7.RowCount
- 1).Cells(p).Value = DataGridView2.Rows(i).Cells(p).Value
        Next
        'Pegando la distancia en la data Resultados Finales

ResultadosFinales.DataGridView8.Rows(ResultadosFinales.DataGridView8.RowCount
- 1).Cells(0).Value = DataGridView3.Rows(i).Cells(0).Value
    Next
    AlgoritmoBusquedaTabu.Show()
End Sub
Private Sub VecinoCercano_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
    Button1.PerformClick()
End Sub
End Class
```

Anexo 10.6. Formulario AlgoritmoBusquedaTabu



Este formulario es llamado *AlgoritmoBusquedaTabu* los controles presentes son: en (1) está un DataGridView llamado *DataGridView1*, en (2) un DataGridView llamado *DataGridView2,* en (3) está un DataGridView llamado *DataGridView3,* en (4) está un DataGridView llamado *DataGridView4,* en (5) está un DataGridView llamado *DataGridView18,* en (6) está un DataGridView llamado *DataGridView8,* en (7) está un DataGridView llamado *DataGridView13,* en (8) está un DataGridView llamado *DataGridView14,* en (9) está un DataGridView llamado *DataGridView15,* en (10) está un DataGridView llamado *DataGridView9,* en (11) está un DataGridView llamado *DataGridView10,* en (12) está un DataGridView llamado *DataGridView5,* en (13) está un DataGridView llamado *DataGridView6,* en (14) está un DataGridView llamado *DataGridView7,* en (15) está un DataGridView llamado *DataGridView11,* en (16) está un DataGridView llamado *DataGridView12,* en (17) está un DataGridView llamado *DataGridView16,* en (18) está un DataGridView llamado *DataGridView17* y en (19) está un Button llamado *Button2.* Los códigos de programación son:

```
Public Class AlgoritmoBusquedaTabu
    Private Sub BusquedaTabuPart1_Load(sender As Object, e As EventArgs)
Handles MyBase.Load
        Button2.PerformClick()
    End Sub
    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click
```

```
        AlgoritmoVecinoCercano.Close()
        'recorrido por filas de las soluciones de VecinoCercano del
formulario Resultados Finales
        For ruta = 0 To ResultadosFinales.DataGridView7.RowCount - 1 '-->
cambiar por ResultadosFinales.DataGridView7.RowCount - 1
            '----------------Colocando todo lo necesario ----
            DataGridView1.ColumnCount =
ResultadosFinales.DataGridView7.ColumnCount
            DataGridView1.RowCount = 1
            DataGridView2.RowCount = 1
            'Mandando la ruta a mejorar de la solucion del VecinoCercano
            For f = 0 To DataGridView1.ColumnCount - 1
                DataGridView1(f, 0).Value =
ResultadosFinales.DataGridView7(f, ruta).Value
            Next
            'Mandando la distancia asociada a la ruta
            DataGridView2(0, 0).Value = ResultadosFinales.DataGridView8(0,
ruta).Value
            'Ajustando la ruta inicial
            For i = 0 To DataGridView1.ColumnCount - 1
                If DataGridView1(i, 0).Value = "0" Then
                    DataGridView1.ColumnCount = i
                    Exit For
                End If
            Next
            If DataGridView1.ColumnCount > 4 Then 'para analizar solo rutas
donde esten mas de 2 destinos
                DataGridView3.RowCount = 1
                DataGridView3.ColumnCount = DataGridView1.ColumnCount
                DataGridView4.RowCount = 1
                'Agregando columnas al data9 o de las permutaciones
                DataGridView9.ColumnCount = DataGridView1.ColumnCount
                'en estas tablas se muestra la solucion inicial obtenida por
el vecino mas cercano
                DataGridView11.ColumnCount = DataGridView1.ColumnCount
                DataGridView11.RowCount = 1
                DataGridView12.RowCount = 1
                For i = 0 To DataGridView11.ColumnCount - 1
                    DataGridView11(i, 0).Value = DataGridView1(i, 0).Value
                Next
                DataGridView12(0, 0).Value = DataGridView2(0, 0).Value
                'en la tabla de solucion de cada iteracion
                DataGridView14.ColumnCount = DataGridView1.ColumnCount
                'en estas tablas se mostrara la solucion final
                DataGridView16.ColumnCount = DataGridView1.ColumnCount
                DataGridView16.RowCount = 1
                'distancia de la solucion final
                DataGridView17.RowCount = 1
                'Cambiando la etiqueta por indices en el data 18
                'columna 0 --> destino
                'columna 1 --> indice
                For i = 1 To DataGridView8.RowCount - 1
                    DataGridView18.Rows.Add()
                    DataGridView18.Rows(DataGridView18.RowCount -
1).Cells(0).Value = DataGridView8(0, i).Value
                    DataGridView18.Rows(DataGridView18.RowCount -
1).Cells(1).Value = i - 1
                Next
                'colocando los indices a las etiquetas del data1 y data8
                'data1
                For i = 0 To DataGridView1.ColumnCount - 1
                    For t = 0 To DataGridView18.RowCount - 1
                        If DataGridView1(i, 0).Value = DataGridView18(0,
t).Value Then
```

```
                            DataGridView1(i, 0).Value = DataGridView18(1,
t).Value
                            Exit For
                        End If
                    Next
                Next

                '----------------Ejecutando la funcion Tabu -----------
                Funcion_tabu()
                ''-------------------Quitando todo para la nueva ruta a
analizar
                'Limpiando
                For i = 0 To DataGridView1.ColumnCount - 1
                    DataGridView1(i, 0).Value = 0
                Next
                DataGridView2(0, 0).Value = 0
                For i = 0 To DataGridView3.ColumnCount - 1
                    DataGridView3(i, 0).Value = 0
                Next
                DataGridView4(0, 0).Value = 0
                'Eliminado filas
                DataGridView5.RowCount = 0
                DataGridView6.RowCount = 0
                DataGridView7.RowCount = 0
                DataGridView9.RowCount = 0
                DataGridView10.RowCount = 0
                For i = 0 To DataGridView11.ColumnCount - 1
                    DataGridView11(i, 0).Value = 0
                Next
                DataGridView12(0, 0).Value = 0
                DataGridView13.RowCount = 0
                DataGridView14.RowCount = 0
                DataGridView15.RowCount = 0
                For i = 0 To DataGridView11.ColumnCount - 1
                    DataGridView16(i, 0).Value = 0
                Next
                DataGridView17(0, 0).Value = 0
                DataGridView18.RowCount = 0
            Else
                '-------------------------------
                'Colocar la solucion del metodo del vecino mas cercano en
form ResultadosFinales de la tabla Tabu
                '-------------------------------
                'Agregando fila al datagrid de tabu de formulario Resultados
Finales
                ResultadosFinales.DataGridView3.Rows.Add()
                ResultadosFinales.DataGridView4.Rows.Add()
                For p = 0 To ResultadosFinales.DataGridView3.ColumnCount - 1

ResultadosFinales.DataGridView3.Rows(ResultadosFinales.DataGridView3.RowCount
- 1).Cells(p).Value = 0
                Next
                'Pegando la ruta en la data Resultados Finales
                For p = 0 To DataGridView1.ColumnCount - 1

ResultadosFinales.DataGridView3.Rows(ResultadosFinales.DataGridView3.RowCount
- 1).Cells(p).Value = DataGridView1.Rows(0).Cells(p).Value
                Next
                'Pegando la distancia en la data Resultados Finales

ResultadosFinales.DataGridView4.Rows(ResultadosFinales.DataGridView4.RowCount
- 1).Cells(0).Value = DataGridView2.Rows(0).Cells(0).Value
                'Limpiando la ruta inicial
                For f = 0 To DataGridView1.ColumnCount - 1
```

```
                    DataGridView1(f, 0).Value = 0
                Next
                DataGridView2(0, 0).Value = 0
            End If

        Next
        ''''''''''''''''''''''''''''''''''''''''''''''''''
        AlgoritmoGenetico.Show()
    End Sub
    Sub Funcion_tabu()
        'Creando una variable para calcular la distancia entre puntos
        Dim Distancia_entre_puntos(0 To DataGridView8.ColumnCount - 1, 0 To
DataGridView8.ColumnCount - 1) As Integer '---------> matriz interna
        For i = 1 To DataGridView8.RowCount - 1 ' Fila
            For j = 1 To DataGridView8.ColumnCount - 1 'Columna
                Distancia_entre_puntos(i - 1, j - 1) = DataGridView8(j,
i).Value
            Next
        Next
        '-------------------------------------------------------------------
-----
        ''''''''''''''''''''''''Realizando la primera iteracion
        '-------------------------------------------------------------------
-----
        '****************************** Permutaciones

'**************************************************************************
        'Identificar los tipos de permutaciones ejemplo: 2 en 2 , 3 en 3 , 4
en 4 , etc.
        Dim tip_permu As Integer = DataGridView1.ColumnCount - 2 '-->
cantidad de solo destinos
        Dim x As Integer = 2 ' Siempre el tipo de permutacion comienza de 2
en 2
        While x < tip_permu
            ' mandarle en que fila se va a permutar
            Dim Num_ciudades As Integer
            Num_ciudades = DataGridView1.ColumnCount - 2
            For i = 1 To Num_ciudades - x + 1
                DataGridView9.Rows.Add()
                'Colocando en la fila creada la ruta inicial para luego
permutar
                For m = 0 To DataGridView1.ColumnCount - 1
                    DataGridView9(m, DataGridView9.RowCount - 1).Value =
DataGridView1(m, 0).Value
                Next
                '****************************
                'Creando un vector donde se juntan las ciudades a permutar --
--> la dimension debe de ser segun el tipo de permutacion (x)
                Dim vector_permutacion(x - 1) As Integer
                For j = i To i + x - 1
                    vector_permutacion(j - i) = DataGridView1(j, 0).Value
                Next
                'Hasta aqui en un vector esta almacenado las ciudades a
permutar
                '****************************
                'Invertir los valores de ese vector
                Dim vector_permutacion_invertido(x - 1) As Integer
                For p = 0 To x - 1
                    vector_permutacion_invertido(p) = vector_permutacion(x -
1 - p)
                Next
                '****************************
                'Colocar ese vector invertido en la ruta
                For j = i To i + x - 1
```

```
                    DataGridView9(j, DataGridView9.RowCount - 1).Value =
vector_permutacion_invertido(j - i)
                Next
            Next
            x = x + 1
        End While
        'Colocando las distancias de cada iteracion
        Dim a1, b1 As Integer
        For k = 0 To DataGridView9.RowCount - 1
            DataGridView10.Rows.Add()
            DataGridView10(0, k).Value = 0
            For j = 0 To DataGridView9.ColumnCount - 2
                a1 = DataGridView9(j, k).Value
                b1 = DataGridView9(j + 1, k).Value
                DataGridView10(0, k).Value = DataGridView10(0, k).Value +
Distancia_entre_puntos(a1, b1)
            Next
        Next
        '*********************************************************************
        '***********  Encontrar la menor solucion de la permutacion ****
        '*********************************************************************
        Dim indice_solucion_iteracion As Integer
        Dim min_distancia
        min_distancia = 1000000
        For i = 0 To DataGridView10.RowCount - 1
            If min_distancia > DataGridView10(0, i).Value Then
                indice_solucion_iteracion = i
                min_distancia = DataGridView10(0, i).Value
            End If
        Next
        indice_solucion_iteracion = indice_solucion_iteracion ' ------>

'*************************************************************************************
        '*** Colocando la solucion de la primera iteracion en data3 para
analizarlo

'*************************************************************************************

        For i = 0 To DataGridView3.ColumnCount - 1
            DataGridView3(i, 0).Value = DataGridView9(i,
indice_solucion_iteracion).Value
        Next
        DataGridView4(0, 0).Value = DataGridView10(0,
indice_solucion_iteracion).Value

'*************************************************************************************
****
        '************** Analizando el A.Eliminado y A.Añadido

'*************************************************************************************
***
        'Analisis de subida
        Dim indice_columna_subida As Integer
        For i = 1 To DataGridView1.ColumnCount - 2
            If DataGridView1(i, 0).Value <> DataGridView3(i, 0).Value Then
                indice_columna_subida = i
                Exit For
            End If
        Next
        indice_columna_subida = indice_columna_subida
        'Analisis de bajada
        Dim indice_columna_bajada As Integer
        For i = DataGridView1.ColumnCount - 2 To 1 Step -1
            If DataGridView1(i, 0).Value <> DataGridView3(i, 0).Value Then
```

```
                indice_columna_bajada = i
                Exit For
            End If
        Next
        indice_columna_bajada = indice_columna_bajada

'*************************************************************************
*
        '******************** Añadiendo A.Eliminado y A.Añadido

'*************************************************************************
*
        'A.Eliminado
        ' Primer arco eliminado --- son del data1
        DataGridView5.Rows.Add()
        DataGridView5.Rows(DataGridView5.RowCount - 1).Cells(0).Value =
DataGridView1(indice_columna_subida - 1, 0).Value
        DataGridView5.Rows(DataGridView5.RowCount - 1).Cells(1).Value =
DataGridView1(indice_columna_subida, 0).Value
        ' Segundo arco eliminado --- son del data1
        DataGridView5.Rows.Add()
        DataGridView5.Rows(DataGridView5.RowCount - 1).Cells(0).Value =
DataGridView1(indice_columna_bajada, 0).Value
        DataGridView5.Rows(DataGridView5.RowCount - 1).Cells(1).Value =
DataGridView1(indice_columna_bajada + 1, 0).Value
        'A.Añadido
        ' Primer arco añadido --- son del data3
        DataGridView6.Rows.Add()
        DataGridView6.Rows(DataGridView6.RowCount - 1).Cells(0).Value =
DataGridView3(indice_columna_subida - 1, 0).Value
        DataGridView6.Rows(DataGridView6.RowCount - 1).Cells(1).Value =
DataGridView3(indice_columna_subida, 0).Value
        ' Segundo arco añadido --- son del data3
        DataGridView6.Rows.Add()
        DataGridView6.Rows(DataGridView6.RowCount - 1).Cells(0).Value =
DataGridView3(indice_columna_bajada, 0).Value
        DataGridView6.Rows(DataGridView6.RowCount - 1).Cells(1).Value =
DataGridView3(indice_columna_bajada + 1, 0).Value

'*************************************************************************
*
        '******************** Añadiendo a la Lista Tabu

'*************************************************************************
*
        'Añadiendo primer A.Añadido a la lista Tabu
        DataGridView7.Rows.Add()
        DataGridView7.Rows(DataGridView7.RowCount - 1).Cells(0).Value =
DataGridView6(0, 0).Value
        DataGridView7.Rows(DataGridView7.RowCount - 1).Cells(1).Value =
DataGridView6(1, 0).Value
        DataGridView7.Rows(DataGridView7.RowCount - 1).Cells(2).Value = 2
        'Añadiendo segundo A.Añadido a la lista Tabu
        DataGridView7.Rows.Add()
        DataGridView7.Rows(DataGridView7.RowCount - 1).Cells(0).Value =
DataGridView6(0, 1).Value
        DataGridView7.Rows(DataGridView7.RowCount - 1).Cells(1).Value =
DataGridView6(1, 1).Value
        DataGridView7.Rows(DataGridView7.RowCount - 1).Cells(2).Value = 2

'*************************************************************************
*
        '******************** Agregando la solucion de la primera iteracion
```

```
'*************************************************************************
*
        'Numero de iteracion --> data 13
        DataGridView13.Rows.Add()
        DataGridView13(0, DataGridView13.RowCount - 1).Value = 1
        'La ruta --> data 14
        DataGridView14.Rows.Add()
        For i = 0 To DataGridView14.ColumnCount - 1
            DataGridView14(i, DataGridView14.RowCount - 1).Value =
DataGridView3(i, 0).Value
        Next
        'Colocando las etiquetas
        For i = 0 To DataGridView14.ColumnCount - 1
            For t = 0 To DataGridView18.RowCount - 1
                If DataGridView14(i, DataGridView14.RowCount - 1).Value =
DataGridView18(1, t).Value Then
                    DataGridView14(i, DataGridView14.RowCount - 1).Value =
DataGridView18(0, t).Value
                    Exit For
                End If
            Next
        Next
        'La distancia
        DataGridView15.Rows.Add()
        DataGridView15(0, DataGridView15.RowCount - 1).Value =
DataGridView4(0, 0).Value

'*************************************************************************
********
        '*** Dejando el formulario listo para una nueva iteracion

'*************************************************************************
*******
        'Pegando la solucion de la iteracion para que sea la solucion inicial
de la siguiente iteracion
        For i = 0 To DataGridView1.ColumnCount - 1
            DataGridView1(i, 0).Value = DataGridView14(i,
DataGridView14.RowCount - 1).Value
        Next
        DataGridView2(0, 0).Value = DataGridView15(0, DataGridView15.RowCount
- 1).Value
        'Limpiando
        For i = 0 To DataGridView3.ColumnCount - 1
            DataGridView3(i, 0).Value = 0
        Next
        DataGridView4(0, 0).Value = 0
        'Eliminado filas
        DataGridView5.RowCount = 0
        DataGridView6.RowCount = 0
        DataGridView9.RowCount = 0
        DataGridView10.RowCount = 0
        'Cambiando las etiquetas
        For i = 0 To DataGridView1.ColumnCount - 1
            For t = 0 To DataGridView18.RowCount - 1
                If DataGridView1(i, 0).Value = DataGridView18(0, t).Value
Then
                    DataGridView1(i, 0).Value = DataGridView18(1, t).Value
                    Exit For
                End If
            Next
        Next
        '************************************************
```

```
        '************** Comenzando para la segunda iteracion hasta la decimo
quinta *
        '*************************************************
        For iteracion = 2 To 15 'Numero total de iteraciones
            Dim cerrar As Integer = 0
            '-----------------------------------------
            'Quitando un tiempo a la lista tabu ---------
            '-----------------------------------------
            For i = 0 To DataGridView7.RowCount - 1
                DataGridView7(2, i).Value = DataGridView7(2, i).Value - 1
            Next
            'Ordenando la lista tabu de mayor a menor
            Dim f1, c1, f2, c2 As Integer
            Dim sube, baja As Double
            For i = 0 To DataGridView7.RowCount - 1
                For j = 0 To DataGridView7.RowCount - 1
                    If DataGridView7(2, i).Value >= DataGridView7(2, j).Value
Then
                        baja = DataGridView7(2, i).Value
                        sube = DataGridView7(2, j).Value
                        f1 = DataGridView7(0, i).Value
                        c1 = DataGridView7(1, i).Value
                        f2 = DataGridView7(0, j).Value
                        c2 = DataGridView7(1, j).Value
                        DataGridView7(2, i).Value = sube  ''''reubicando
                        DataGridView7(0, i).Value = f2
                        DataGridView7(1, i).Value = c2
                        DataGridView7(2, j).Value = baja    ''''reubicando
                        DataGridView7(0, j).Value = f1
                        DataGridView7(1, j).Value = c1
                    End If
                Next
            Next
            'Quitando de la lista tabu los de tiempo cero
            For i = 0 To DataGridView7.RowCount - 1
                If DataGridView7(2, i).Value = 0 Then
                    DataGridView7.RowCount = i
                    Exit For
                End If
            Next
            '----------------------
            'Permutaciones ---------
            '----------------------
            tip_permu = DataGridView1.ColumnCount - 2 '--> cantidad de solo
destinos
            x = 2 ' Siempre el tipo de permutacion comienza de 2 en 2
            While x < tip_permu
                'mandarle en que fila se va a permutar
                Dim Num_ciudades As Integer
                Num_ciudades = DataGridView1.ColumnCount - 2
                For i = 1 To Num_ciudades - x + 1
                    DataGridView9.Rows.Add()
                    'Colocando en la fila creada la ruta inicial para luego
permutar
                    For m = 0 To DataGridView1.ColumnCount - 1
                        DataGridView9(m, DataGridView9.RowCount - 1).Value =
DataGridView1(m, 0).Value
                    Next
                    '******************************
                    'Creando un vector donde se juntan las ciudades a
permutar ----> la dimension debe de ser segun el tipo de permutacion (x)
                    Dim vector_permutacion(x - 1) As Integer
                    For j = i To i + x - 1
                        vector_permutacion(j - i) = DataGridView1(j, 0).Value
```

```
                    Next
                    'Hasta aqui en un vector esta almacenado las ciudades a
permutar
                    '***************************
                    'Invertir los valores de ese vector
                    Dim vector_permutacion_invertido(x - 1) As Integer
                    For p = 0 To x - 1
                        vector_permutacion_invertido(p) =
vector_permutacion(x - 1 - p)
                    Next
                    '********************
                    'Colocar ese vector invertido en la ruta
                    For j = i To i + x - 1
                        DataGridView9(j, DataGridView9.RowCount - 1).Value =
vector_permutacion_invertido(j - i)
                    Next
                Next
                x = x + 1
            End While
            'Colocando las distancias de cada iteracion
            For k = 0 To DataGridView9.RowCount - 1
                DataGridView10.Rows.Add()
                DataGridView10(0, k).Value = 0
                For j = 0 To DataGridView9.ColumnCount - 2
                    a1 = DataGridView9(j, k).Value
                    b1 = DataGridView9(j + 1, k).Value
                    DataGridView10(0, k).Value = DataGridView10(0, k).Value +
Distancia_entre_puntos(a1, b1)
                Next
            Next
            Dim ite
            ite = iteracion
            While ite = iteracion
                '-------------------------------------------------
                'Ver si hay soluciones en la iteracion ---------
                '-------------------------------------------------
                If DataGridView9.RowCount = 0 Then
                    cerrar = 1
                    ' MsgBox("fin de la permutacion por que en la iteracion
ya no tiene mas soluciones")
                    Exit While
                End If
                If DataGridView9.RowCount <> 0 Then
                    'Encontrar la mejor solucion de la iteracion
                    Dim indi_solucion_iteracion As Integer
                    min_distancia = 1000000
                    For i = 0 To DataGridView10.RowCount - 1
                        If min_distancia > DataGridView10(0, i).Value Then
                            indi_solucion_iteracion = i
                            min_distancia = DataGridView10(0, i).Value
                        End If
                    Next
                    indi_solucion_iteracion = indi_solucion_iteracion

'*************************************************************************
                    '*** Colocando la solucion de la primera iteracion en
data3 para analizarlo

'*************************************************************************
                    For i = 0 To DataGridView3.ColumnCount - 1
                        DataGridView3(i, 0).Value = DataGridView9(i,
indi_solucion_iteracion).Value
                    Next
```

```
                    DataGridView4(0, 0).Value = DataGridView10(0,
indi_solucion_iteracion).Value
                '*****************************************************
                    '************** Analizando el A.Eliminado y A.Añadido
                    '***************************
                    'Analisis de subida
                    Dim indi_columna_subida As Integer
                    For i = 1 To DataGridView1.ColumnCount - 2
                        If DataGridView1(i, 0).Value <> DataGridView3(i,
0).Value Then
                            indi_columna_subida = i
                            Exit For
                        End If
                    Next
                    indi_columna_subida = indi_columna_subida
                    'Analisis de bajada
                    Dim indi_columna_bajada As Integer
                    For i = DataGridView1.ColumnCount - 2 To 1 Step -1
                        If DataGridView1(i, 0).Value <> DataGridView3(i,
0).Value Then
                            indi_columna_bajada = i
                            Exit For
                        End If
                    Next
                    indi_columna_bajada = indi_columna_bajada

                    *****************************************************
                    '******************** Añadiendo A.Eliminado y A.Añadido
                    '************************************
                    'A.Eliminado
                    ' Primer arco eliminado --- son del data1
                    DataGridView5.Rows.Add()
                    DataGridView5.Rows(DataGridView5.RowCount -
1).Cells(0).Value = DataGridView1(indi_columna_subida - 1, 0).Value
                    DataGridView5.Rows(DataGridView5.RowCount -
1).Cells(1).Value = DataGridView1(indi_columna_subida, 0).Value
                    ' Segundo arco eliminado --- son del data1
                    DataGridView5.Rows.Add()
                    DataGridView5.Rows(DataGridView5.RowCount -
1).Cells(0).Value = DataGridView1(indi_columna_bajada, 0).Value
                    DataGridView5.Rows(DataGridView5.RowCount -
1).Cells(1).Value = DataGridView1(indi_columna_bajada + 1, 0).Value
                    'A.Añadido
                    ' Primer arco añadido --- son del data3
                    DataGridView6.Rows.Add()
                    DataGridView6.Rows(DataGridView6.RowCount -
1).Cells(0).Value = DataGridView3(indi_columna_subida - 1, 0).Value
                    DataGridView6.Rows(DataGridView6.RowCount -
1).Cells(1).Value = DataGridView3(indi_columna_subida, 0).Value
                    ' Segundo arco añadido --- son del data3
                    DataGridView6.Rows.Add()
                    DataGridView6.Rows(DataGridView6.RowCount -
1).Cells(0).Value = DataGridView3(indi_columna_bajada, 0).Value
                    DataGridView6.Rows(DataGridView6.RowCount -
1).Cells(1).Value = DataGridView3(indi_columna_bajada + 1, 0).Value
                    '*********************************
                    '******************** Viendo si los A.Eliminados pertecen
a la lista tabu
                    '************************
                    'Tiene que estar los 2 arcos eliminados para ser
considerado tabu
                    Dim resp1 As Integer = 0
                    Dim resp2 As Integer = 0
                    For t = 0 To DataGridView7.RowCount - 1
```

```
                        If DataGridView5(0, 0).Value = DataGridView7(0,
t).Value And DataGridView5(1, 0).Value = DataGridView7(1, t).Value Then
                            resp1 = 1
                            Exit For
                        End If
                    Next
                    resp1 = resp1
                    For t = 0 To DataGridView7.RowCount - 1
                        If DataGridView5(0, 1).Value = DataGridView7(0,
t).Value And DataGridView5(1, 1).Value = DataGridView7(1, t).Value Then
                            resp2 = 1
                            Exit For
                        End If
                    Next
                    resp2 = resp2
                    Dim respuesta_tabu As Integer
                    respuesta_tabu = resp1 + resp2
                    'respuesta_tabu = 2 ----> si estan en la lista tabu
                    If respuesta_tabu = 1 Or respuesta_tabu = 0 Then 'la
solucion no esta en la lista tabu
                        '----------------------------
                        '----------Continuando
                        '----------------------------
                        ********************************
                        '******************** Añadiendo a la Lista Tabu
                        'Añadiendo primer A.Añadido a la lista Tabu
                        DataGridView7.Rows.Add()
                        DataGridView7.Rows(DataGridView7.RowCount -
1).Cells(0).Value = DataGridView6(0, 0).Value
                        DataGridView7.Rows(DataGridView7.RowCount -
1).Cells(1).Value = DataGridView6(1, 0).Value
                        DataGridView7.Rows(DataGridView7.RowCount -
1).Cells(2).Value = 2

                        'Añadiendo segundo A.Añadido a la lista Tabu
                        DataGridView7.Rows.Add()
                        DataGridView7.Rows(DataGridView7.RowCount -
1).Cells(0).Value = DataGridView6(0, 1).Value
                        DataGridView7.Rows(DataGridView7.RowCount -
1).Cells(1).Value = DataGridView6(1, 1).Value
                        DataGridView7.Rows(DataGridView7.RowCount -
1).Cells(2).Value = 2
                        **********************************
                        '******************** Agregando la solucion la
iteracion
                        '****************************
                        'Numero de iteracion --> data 13
                        DataGridView13.Rows.Add()
                        DataGridView13(0, DataGridView13.RowCount - 1).Value
= iteracion
                        'La ruta --> data 14
                        DataGridView14.Rows.Add()
                        For i = 0 To DataGridView14.ColumnCount - 1
                            DataGridView14(i, DataGridView14.RowCount -
1).Value = DataGridView3(i, 0).Value
                        Next
                        'Colocando las etiquetas
                        For i = 0 To DataGridView14.ColumnCount - 1
                            For t = 0 To DataGridView18.RowCount - 1
                                If DataGridView14(i, DataGridView14.RowCount
- 1).Value = DataGridView18(1, t).Value Then
                                    DataGridView14(i, DataGridView14.RowCount
- 1).Value = DataGridView18(0, t).Value
                                    Exit For
```

```
                            End If
                        Next
                    Next
                    'La distancia
                    DataGridView15.Rows.Add()
                    DataGridView15(0, DataGridView15.RowCount - 1).Value
= DataGridView4(0, 0).Value
                    '**********************************
                    '*** Dejando el formulario listo para una nueva
iteracion
                    '******************************
                    'Pegando la solucion de la iteracion para que sea la
solucion inicial de la siguiente iteracion
                    For i = 0 To DataGridView1.ColumnCount - 1
                        DataGridView1(i, 0).Value = DataGridView14(i,
DataGridView14.RowCount - 1).Value
                    Next
                    DataGridView2(0, 0).Value = DataGridView15(0,
DataGridView15.RowCount - 1).Value
                    'MsgBox("Solucion de la iteracion" +
iteracion.ToString)
                    'Limpiando
                    For i = 0 To DataGridView3.ColumnCount - 1
                        DataGridView3(i, 0).Value = 0
                    Next
                    DataGridView4(0, 0).Value = 0
                    'Eliminado filas
                    DataGridView5.RowCount = 0
                    DataGridView6.RowCount = 0
                    DataGridView9.RowCount = 0
                    DataGridView10.RowCount = 0
                    'Cambiando las etiquetas
                    For i = 0 To DataGridView1.ColumnCount - 1
                        For t = 0 To DataGridView18.RowCount - 1
                            If DataGridView1(i, 0).Value =
DataGridView18(0, t).Value Then
                                DataGridView1(i, 0).Value =
DataGridView18(1, t).Value
                                Exit For
                            End If
                        Next
                    Next
                    Exit While 'Se pasa a la siguiente iteracion
                ElseIf respuesta_tabu = 2 Then 'la solucion si esta en la
lista tabu
                    ************************
                    '*** Analizar si la solucion mejora la solucion de la
iteracion anterior
                    '**********************************
                    Dim solu_ite_anterior As Integer = DataGridView2(0,
0).Value
                    Dim solu_ite_actual As Integer = DataGridView4(0,
0).Value
                    If solu_ite_actual < solu_ite_anterior Then ' la
solucion tabu es mejor
                        '----------------------------
                        '----------Continuando
                        '----------------------------
                        ************************************
                        '******************** Añadiendo a la Lista Tabu
                        '************************************
                        'Añadiendo primer A.Añadido a la lista Tabu
                        DataGridView7.Rows.Add()
```

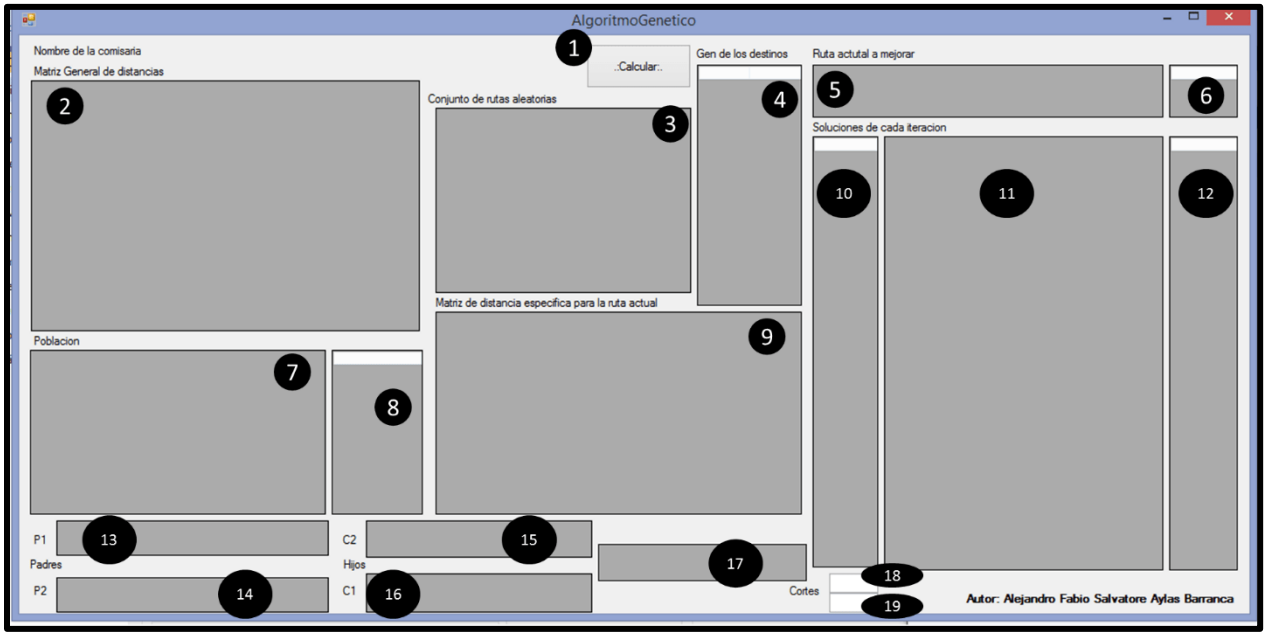
```
                            DataGridView7.Rows(DataGridView7.RowCount -
1).Cells(0).Value = DataGridView6(0, 0).Value
                            DataGridView7.Rows(DataGridView7.RowCount -
1).Cells(1).Value = DataGridView6(1, 0).Value
                            DataGridView7.Rows(DataGridView7.RowCount -
1).Cells(2).Value = 2
                            'Añadiendo segundo A.Añadido a la lista Tabu
                            DataGridView7.Rows.Add()
                            DataGridView7.Rows(DataGridView7.RowCount -
1).Cells(0).Value = DataGridView6(0, 1).Value
                            DataGridView7.Rows(DataGridView7.RowCount -
1).Cells(1).Value = DataGridView6(1, 1).Value
                            DataGridView7.Rows(DataGridView7.RowCount -
1).Cells(2).Value = 2
                            '******************** Agregando la solucion la
iteracion
                            '*************************
                            'Numero de iteracion --> data 13
                            DataGridView13.Rows.Add()
                            DataGridView13(0, DataGridView13.RowCount -
1).Value = iteracion
                            'La ruta --> data 14
                            DataGridView14.Rows.Add()
                            For i = 0 To DataGridView14.ColumnCount - 1
                                DataGridView14(i, DataGridView14.RowCount -
1).Value = DataGridView3(i, 0).Value
                            Next
                            'Colocando las etiquetas
                            For i = 0 To DataGridView14.ColumnCount - 1
                                For t = 0 To DataGridView18.RowCount - 1
                                    If DataGridView14(i,
DataGridView14.RowCount - 1).Value = DataGridView18(1, t).Value Then
                                        DataGridView14(i,
DataGridView14.RowCount - 1).Value = DataGridView18(0, t).Value
                                        Exit For
                                    End If
                                Next
                            Next
                            'La distancia
                            DataGridView15.Rows.Add()
                            DataGridView15(0, DataGridView15.RowCount -
1).Value = DataGridView4(0, 0).Value
                        '*************************************
                            '*** Dejando el formulario listo para una nueva
iteracion
                            '*********************************
                            'Pegando la solucion de la iteracion para que sea
la solucion inicial de la siguiente iteracion
                            For i = 0 To DataGridView1.ColumnCount - 1
                                DataGridView1(i, 0).Value = DataGridView14(i,
DataGridView14.RowCount - 1).Value
                            Next
                            DataGridView2(0, 0).Value = DataGridView15(0,
DataGridView15.RowCount - 1).Value
                            'MsgBox("Solucion de la iteracion" +
iteracion.ToString)
                            'Limpiando
                            For i = 0 To DataGridView3.ColumnCount - 1
                                DataGridView3(i, 0).Value = 0
                            Next
                            DataGridView4(0, 0).Value = 0
                            'Eliminado filas
                            DataGridView5.RowCount = 0
                            DataGridView6.RowCount = 0
```

```
                            DataGridView9.RowCount = 0
                            DataGridView10.RowCount = 0
                           'Cambiando las etiquetas
                            For i = 0 To DataGridView1.ColumnCount - 1
                                For t = 0 To DataGridView18.RowCount - 1
                                    If DataGridView1(i, 0).Value =
DataGridView18(0, t).Value Then
                                        DataGridView1(i, 0).Value =
DataGridView18(1, t).Value
                                        Exit For
                                    End If
                                Next
                            Next
                            Exit While 'Se pasa a la siguiente iteracion

                        ElseIf solu_ite_actual >= solu_ite_anterior Then ' la
solucion tabu no es mejor
                            '-----------------------------------------------
---
                            '---------- Buscar otra solucion en las
iteraciones
                            '-----------------------------------------------
--
                            'Eliminar esta solucion indi_solucion_iteracion
de la fila del data9 y data10
                            If indi_solucion_iteracion <>
DataGridView10.RowCount - 1 Then 'si la mala solucion no esta en la ultima
fila
                                'Reemplazandolo por el ultimo valor de la
fila
                                Dim v_bajar
                                Dim v_subir
                                'Cambiando la solucion de la iteracion
                                For i = 0 To DataGridView9.ColumnCount - 1
                                    v_bajar = DataGridView9(i,
indi_solucion_iteracion).Value
                                    v_subir = DataGridView9(i,
DataGridView9.RowCount - 1).Value

                                    DataGridView9(i,
indi_solucion_iteracion).Value = v_subir
                                    DataGridView9(i, DataGridView9.RowCount -
1).Value = v_bajar
                                Next
                                'Cambiando la solucion de la distancia
                                v_bajar = DataGridView10(0,
indi_solucion_iteracion).Value
                                v_subir = DataGridView10(0,
DataGridView10.RowCount - 1).Value
                                DataGridView10(0,
indi_solucion_iteracion).Value = v_subir
                                DataGridView10(0, DataGridView10.RowCount -
1).Value = v_bajar
                                'Cortando la ultima fila
                                indi_solucion_iteracion =
DataGridView9.RowCount - 1
                                DataGridView9.RowCount =
indi_solucion_iteracion
                                DataGridView10.RowCount =
indi_solucion_iteracion
                                'MsgBox("corte1")
                            End If
```

```
                            If indi_solucion_iteracion =
DataGridView10.RowCount - 1 Then 'si la mala solucion si esta en la ultima
fila
                                'cortar la solucion
                                DataGridView9.RowCount =
indi_solucion_iteracion
                                DataGridView10.RowCount =
indi_solucion_iteracion
                                'MsgBox("corte2")
                            End If
                            'Limpiando
                            For i = 0 To DataGridView3.ColumnCount - 1
                                DataGridView3(i, 0).Value = 0
                            Next
                            DataGridView4(0, 0).Value = 0
                            'Eliminado filas
                            DataGridView5.RowCount = 0
                            DataGridView6.RowCount = 0
                        End If
                    End If
                End If
            End While
            If cerrar = 1 Then
                Exit For
            End If
        Next
        '*********************************************************
        '************** Buscar la mejor solucion  de las iteraciones
        '*********************************************************
        'Buscando la mejor solucion de las iteraciones
        Dim solu_final As Integer = DataGridView15(0, 0).Value
        Dim indi_solu_final As Integer = 0
        For i = 0 To DataGridView15.RowCount - 1
            If solu_final > DataGridView15(0, i).Value Then
                indi_solu_final = i
                solu_final = DataGridView15(0, i).Value
            End If
        Next
        indi_solu_final = indi_solu_final
        solu_final = solu_final
        'Verificando si la solucion encontrada es mejor que la solucion del
algoritmo del metodo del vecino mas cercano
        If solu_final <= DataGridView12(0, 0).Value Then
            For i = 0 To DataGridView14.ColumnCount - 1
                DataGridView16(i, 0).Value = DataGridView14(i,
indi_solu_final).Value
            Next
            DataGridView17(0, 0).Value = solu_final
        ElseIf solu_final > DataGridView12(0, 0).Value Then
            For i = 0 To DataGridView14.ColumnCount - 1
                DataGridView16(i, 0).Value = DataGridView12(i, 0).Value
            Next
            DataGridView17(0, 0).Value = DataGridView12(0, 0).Value
        End If
        '*********************************************************
        '******* Asignando la mejor solucion al form ResultadosFinales
*******
        '*********************************************************
        ResultadosFinales.DataGridView3.Rows.Add()
        For i = 0 To ResultadosFinales.DataGridView3.ColumnCount - 1
            ResultadosFinales.DataGridView3(i,
ResultadosFinales.DataGridView3.RowCount - 1).Value = 0
        Next
        For i = 0 To DataGridView16.ColumnCount - 1
```

```
            ResultadosFinales.DataGridView3(i,
ResultadosFinales.DataGridView3.RowCount - 1).Value = DataGridView16(i,
0).Value
        Next
        'distancia
        ResultadosFinales.DataGridView4.Rows.Add()
        ResultadosFinales.DataGridView4(0,
ResultadosFinales.DataGridView3.RowCount - 1).Value = DataGridView17(0,
0).Value
    End Sub
End Class
```

Anexo 10.7. Formulario AlgoritmoGenetico



Este formulario es llamado *AlgoritmoGenetico* los controles presentes son: en (1) está un Boton llamado *Button2*, en (2) un DataGridView llamado *DataGridView9,* en (3) está un DataGridView llamado *DataGridView10,* en (4) se encuentra un DataGridView llamado *DataGridView8,* en (5) un DataGridView llamado *DataGridView1,* en (6) un DataGridView llamado *DataGridView2,* en (7) un DataGridView llamado *DataGridView3,* en (8) un DataGridView llamado *DataGridView4,* en (9) un DataGridView llamado *DataGridView7,* en (10) un DataGridView llamado *DataGridView20,* en (11) un DataGridView llamado *DataGridView21,* en (12) un DataGridView llamado *DataGridView22,* en (13) un DataGridView llamado *DataGridView5,* en (14) un DataGridView llamado *DataGridView6,* en (15) un DataGridView llamado *DataGridView11,* en (16) un DataGridView llamado *DataGridView12,* en (17) un

DataGridView llamado *DataGridView14,* en (18) un TextBox llamado *TextBox1* y en (19) está un TextBox llamado *TextBox2.* Los códigos de programación son:

```
Public Class AlgoritmoGenetico
    Dim capacidad As Integer = Comisaria.NumPuntos.Text '---> numero maximo
de destinos para visitar por ruta
    Private Sub AlgoritmoGenetico_Load(sender As Object, e As EventArgs)
Handles MyBase.Load
        Button2.PerformClick()
    End Sub
    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click
        AlgoritmoBusquedaTabu.Close()
        'Creacion de la tabla de conjunto de rutas iniciales
        DataGridView10.ColumnCount = capacidad + 2
        'Tabla de indices que se va a sortear
        For i = 2 To DataGridView9.RowCount - 1
            DataGridView8.Rows.Add()
            DataGridView8(0, DataGridView8.RowCount - 1).Value =
DataGridView9(i, 0).Value
        Next
        'Creacion de rutas aleatorias
        Dim destino_sorteado As Integer
        While DataGridView8.RowCount <> 0
            DataGridView10.Rows.Add()
            For i = 0 To DataGridView10.ColumnCount - 1
                DataGridView10(i, DataGridView10.RowCount - 1).Value = 0
            Next
            DataGridView10(0, DataGridView10.RowCount - 1).Value =
DataGridView9(0, 1).Value
            For i = 1 To capacidad ' -> max. puntos por ruta
                If DataGridView8.RowCount = 0 Then
                    Exit For
                Else
                    Randomize()
                    destino_sorteado = CInt(Int(((DataGridView8.RowCount - 1)
* Rnd()) + 0))
                    'Buscando columna donde colocar
                    For j = 0 To DataGridView10.ColumnCount - 1
                        If DataGridView10(j, DataGridView10.RowCount -
1).Value = 0.ToString Then
                            DataGridView10(j, DataGridView10.RowCount -
1).Value = DataGridView8(0, destino_sorteado).Value
                            Exit For
                        End If
                    Next
                    'Actualizando tabla
                    Dim su1, ba1
                    su1 = DataGridView8(0, DataGridView8.RowCount - 1).Value
                    ba1 = DataGridView8(0, destino_sorteado).Value
                    DataGridView8(0, DataGridView8.RowCount - 1).Value = ba1
                    DataGridView8(0, destino_sorteado).Value = su1
                    DataGridView8.RowCount = DataGridView8.RowCount - 1
                End If
            Next
        End While
        'Colocando el destino final en cada ruta
        For i = 0 To DataGridView10.RowCount - 1
            For j = 0 To DataGridView10.ColumnCount - 1
                If DataGridView10(j, i).Value = 0.ToString Then
                    DataGridView10(j, i).Value = DataGridView9(0, 1).Value
                    Exit For
                End If
```

```
            Next
        Next
        '''''''
```

```
                    'Tabla de los Padres
                    DataGridView5.ColumnCount = DataGridView1.ColumnCount - 1
                    DataGridView5.RowCount = 1
                    DataGridView6.ColumnCount = DataGridView1.ColumnCount - 1
                    DataGridView6.RowCount = 1
                    'Tabla de Hijos
                    'en los C2 y C1 -- > los hijos
                    DataGridView11.ColumnCount = DataGridView1.ColumnCount -
1
                    DataGridView11.RowCount = 1
                    DataGridView12.ColumnCount = DataGridView1.ColumnCount -
1
                    DataGridView12.RowCount = 1
                    'Tabla de apoyo para cruces
                    DataGridView14.ColumnCount = DataGridView5.ColumnCount
                    DataGridView14.RowCount = 1
                    'Colocando la solucion inicial en la primera fila de la
tabla de iteraciones
                    DataGridView20.Rows.Add()
                    DataGridView21.Rows.Add()
                    DataGridView22.Rows.Add()
                    DataGridView20(0, DataGridView20.RowCount - 1).Value = 0
                    For i = 0 To DataGridView1.ColumnCount - 1
                        DataGridView21(i, DataGridView21.RowCount - 1).Value
= DataGridView1(i, 0).Value
                    Next
                    DataGridView22(0, DataGridView22.RowCount - 1).Value =
DataGridView2(0, 0).Value
                    'Creando la matriz de distancia asociada a la ruta actual
                    Dim num_destinos_ruta As Integer =
DataGridView1.ColumnCount - 1
                    DataGridView7.ColumnCount = num_destinos_ruta + 1
                    DataGridView7.RowCount = num_destinos_ruta + 1
                    For i = 1 To DataGridView7.ColumnCount - 1
                        DataGridView7(i, 0).Value = DataGridView1(i - 1,
0).Value

                        DataGridView7(0, i).Value = DataGridView1(i - 1,
0).Value
                    Next
                    Dim a, b
                    Dim fil_a As Integer
                    Dim col_b As Integer
                    For i = 1 To DataGridView7.RowCount - 1
                        a = DataGridView7(0, i).Value
                        'Buscando "a" en matrizGeneral de distancias
                        For f = 1 To DataGridView9.RowCount - 1
                            If a = DataGridView9(0, f).Value Then
                                fil_a = f
                                Exit For
                            End If
                        Next
                        fil_a = fil_a
                        For j = 1 To DataGridView7.ColumnCount - 1
                            b = DataGridView7(j, 0).Value
                            'Buscando "b" en matrizGeneral de distancias
                            For f = 1 To DataGridView9.ColumnCount - 1
                                If b = DataGridView9(f, 0).Value Then
                                    col_b = f
                                    Exit For
                                End If
                            Next
                            col_b = col_b
                            'Colocando la distancia
```

```
                        DataGridView7(j, i).Value = DataGridView9(col_b,
fil_a).Value
                    Next
                Next
                'Tabla de Gen de los destinos
                'En el data 18
                ' columna 0 - -> destino
                'columna 1 - -> indice genetico
                For i = 1 To DataGridView7.RowCount - 1 'recorrido por
filas de la matriz
                    DataGridView8.Rows.Add()
                    DataGridView8.Rows(DataGridView8.RowCount -
1).Cells(0).Value = DataGridView7(0, i).Value
                    DataGridView8.Rows(DataGridView8.RowCount -
1).Cells(1).Value = i - 1
                Next
                '----------------------- Ejecutar Algoritmo genetico ----
-----------------
                Operacion_genetico()
                '----------------------- Buscando la mejor solucion de las
400 iteraciones
                Dim solu_final As Integer = DataGridView22(0, 0).Value
                Dim indi_solu_final As Integer = 0
                For i = 0 To DataGridView22.RowCount - 1
                    If solu_final > DataGridView22(0, i).Value Then
                        indi_solu_final = i
                        solu_final = DataGridView22(0, i).Value
                    End If
                Next
                indi_solu_final = indi_solu_final
                solu_final = solu_final
                '---------- Colocando la solucion de las 400 iteraciones
en la corrida
                SolucionesAG.DataGridView1.Rows.Add()
                SolucionesAG.DataGridView2.Rows.Add()
                For i = 0 To SolucionesAG.DataGridView1.ColumnCount - 1
                    SolucionesAG.DataGridView1(i,
SolucionesAG.DataGridView1.RowCount - 1).Value = 0
                Next
                For i = 0 To DataGridView21.ColumnCount - 1
                    SolucionesAG.DataGridView1(i,
SolucionesAG.DataGridView1.RowCount - 1).Value = DataGridView21(i,
indi_solu_final).Value
                Next
                'distancia
                SolucionesAG.DataGridView2(0,
SolucionesAG.DataGridView2.RowCount - 1).Value = solu_final
                ''----------------- Dejar todo limpio para una nueva
iteracion -----------
                'Eliminado filas
                DataGridView3.RowCount = 0
                DataGridView4.RowCount = 0
                DataGridView8.RowCount = 0
                DataGridView20.RowCount = 0
                DataGridView21.RowCount = 0
                DataGridView22.RowCount = 0
                For i = 0 To DataGridView5.ColumnCount - 1
                    DataGridView5(i, 0).Value = 0
                    DataGridView6(i, 0).Value = 0
                    DataGridView11(i, 0).Value = 0
                    DataGridView12(i, 0).Value = 0
                    DataGridView14(i, 0).Value = 0
                Next
                'Matriz especifica
```

```
                    DataGridView7.RowCount = 0
                    DataGridView7.ColumnCount = 0
                Next
                '*******************************************************
                ' Asignando la mejor solucion de los  intentos al form
ResultadosFinales
                Dim sol_final As Integer
                sol_final = SolucionesAG.DataGridView2(0, 0).Value
                Dim indi_sol_final As Integer = 0
                For i = 0 To SolucionesAG.DataGridView2.RowCount - 1
                    If sol_final > SolucionesAG.DataGridView2(0, i).Value
Then
                        indi_sol_final = i
                        sol_final = SolucionesAG.DataGridView2(0, i).Value
                    End If
                Next
                indi_sol_final = indi_sol_final
                sol_final = sol_final
                '*******************************************************
                ' Asignando la mejor solucion al form ResultadosFinales
                ResultadosFinales.DataGridView5.Rows.Add()
                For i = 0 To ResultadosFinales.DataGridView5.ColumnCount - 1
                    ResultadosFinales.DataGridView5(i,
ResultadosFinales.DataGridView5.RowCount - 1).Value = 0
                Next
                For i = 0 To SolucionesAG.DataGridView1.ColumnCount - 1
                    ResultadosFinales.DataGridView5(i,
ResultadosFinales.DataGridView5.RowCount - 1).Value =
SolucionesAG.DataGridView1(i, indi_sol_final).Value
                Next
                'distancia
                ResultadosFinales.DataGridView6.Rows.Add()
                ResultadosFinales.DataGridView6(0,
ResultadosFinales.DataGridView6.RowCount - 1).Value = sol_final
                '''Dejando limpio el formulario de apoyo para una nueva ruta
                SolucionesAG.DataGridView1.ColumnCount = 0
                SolucionesAG.DataGridView1.RowCount = 0
                SolucionesAG.DataGridView2.RowCount = 0
                '''------------------ Dejar todo limpio para evaluar una
nueva ruta -----------
                'Eliminado filas
                DataGridView3.RowCount = 0
                DataGridView4.RowCount = 0
                DataGridView8.RowCount = 0
                DataGridView20.RowCount = 0
                DataGridView21.RowCount = 0
                DataGridView22.RowCount = 0
                For i = 0 To DataGridView1.ColumnCount - 1
                    DataGridView1(i, 0).Value = 0
                Next
                DataGridView2(0, 0).Value = 0
                For i = 0 To DataGridView5.ColumnCount - 1
                    DataGridView5(i, 0).Value = 0
                    DataGridView6(i, 0).Value = 0
                    DataGridView11(i, 0).Value = 0
                    DataGridView12(i, 0).Value = 0
                    DataGridView14(i, 0).Value = 0
                Next
                'Matriz especifica
                DataGridView7.RowCount = 0
                DataGridView7.ColumnCount = 0
            Else
                '-----------------------------------
                'Colocar la solucion del metodo ahorro a Resultados Finales
```

```
                ResultadosFinales.DataGridView5.Rows.Add()
                ResultadosFinales.DataGridView6.Rows.Add()
                For i = 0 To ResultadosFinales.DataGridView5.ColumnCount - 1
                    ResultadosFinales.DataGridView5(i,
ResultadosFinales.DataGridView5.RowCount - 1).Value = 0
                Next
                'Colocando la ruta
                For i = 0 To DataGridView1.ColumnCount - 1
                    ResultadosFinales.DataGridView5(i,
ResultadosFinales.DataGridView5.RowCount - 1).Value = DataGridView1(i,
0).Value
                Next
                'Colocando la distancia
                ResultadosFinales.DataGridView6(0,
ResultadosFinales.DataGridView6.RowCount - 1).Value = DataGridView2(0,
0).Value
                'Limpiando la ruta inicial
                For f = 0 To DataGridView1.ColumnCount - 1
                    DataGridView1(f, 0).Value = 0
                Next
                DataGridView2(0, 0).Value = 0
            End If
        Next
        ResultadosFinales.Show()
    End Sub
    Sub Operacion_genetico()
        'Creando una variable para calcular la distancia entre puntos
        '-> matriz interna
        Dim Distancia_entre_puntos(0 To DataGridView7.ColumnCount - 1, 0 To
DataGridView7.ColumnCount - 1) As Integer
        For i = 1 To DataGridView7.RowCount - 1 ' Fila
            For j = 1 To DataGridView7.ColumnCount - 1 'Columna
                Distancia_entre_puntos(i - 1, j - 1) = DataGridView7(j,
i).Value
            Next
        Next
        '----------------------------------------
        ' Generando la poblacion inicial
        '----------------------------------------
        'Seleccion aleatoria de posicion de fila de tabla Gen
        Dim posicion_sorteada As Integer
        For i = 1 To 6 'tamaño de la poblacion
            DataGridView3.Rows.Add()
            DataGridView3(0, DataGridView3.RowCount - 1).Value = 0
            Randomize() 'Los aleatorias cambian cada vez que se ejecute el
programa
            For p = 1 To DataGridView1.ColumnCount - 2
                'CInt(Int((6 * Rnd()) + 1)) --> entre 1 y 6
                posicion_sorteada = CInt(Int(((DataGridView8.RowCount - 1) *
Rnd()) + 1))
                'Añadiendo a la ruta de la poblacion
                DataGridView3(p, DataGridView3.RowCount - 1).Value =
DataGridView8(1, posicion_sorteada).Value
                'Actualizando tabla de Gen
                Dim sub1, sub2, baj1, baj2
                sub1 = DataGridView8(0, DataGridView8.RowCount - 1).Value
                sub2 = DataGridView8(1, DataGridView8.RowCount - 1).Value
                baj1 = DataGridView8(0, posicion_sorteada).Value
                baj2 = DataGridView8(0, posicion_sorteada).Value
                DataGridView8(0, DataGridView8.RowCount - 1).Value = baj1
                DataGridView8(1, DataGridView8.RowCount - 1).Value = baj2
                DataGridView8(0, posicion_sorteada).Value = sub1
                DataGridView8(1, posicion_sorteada).Value = sub2
                DataGridView8.RowCount = DataGridView8.RowCount - 1
```

```
            Next
            'Recuperando la tabla de Gen
            DataGridView8.RowCount = 0
            For y = 1 To DataGridView7.RowCount - 1
                DataGridView8.Rows.Add()
                DataGridView8.Rows(DataGridView8.RowCount - 1).Cells(0).Value
= DataGridView7(0, y).Value
                DataGridView8.Rows(DataGridView8.RowCount - 1).Cells(1).Value
= y - 1
            Next
            'Colocando el punto final
            DataGridView3(DataGridView3.ColumnCount - 1,
DataGridView3.RowCount - 1).Value = DataGridView3(0, DataGridView3.RowCount -
1).Value
        Next
        'Colocando sus distancias correspondientes a cada ruta
        Dim a1, b1 As Integer
        For k = 0 To DataGridView3.RowCount - 1
            DataGridView4.Rows.Add()
            DataGridView4(0, k).Value = 0
            For j = 0 To DataGridView3.ColumnCount - 2
                a1 = DataGridView3(j, k).Value
                b1 = DataGridView3(j + 1, k).Value
                DataGridView4(0, k).Value = DataGridView4(0, k).Value +
Distancia_entre_puntos(a1, b1)
            Next
        Next
        ''''''''''''
        'Iteraciones
        For iteracion = 1 To 400 'cambiar por 400
            '--------------------------------------
            ' Buscando los 2 padres de la poblacion
            '--------------------------------------
            Dim indice_solucion_iteracion As Integer
            Dim min_distancia = 1000000
            For i = 0 To DataGridView4.RowCount - 1
                If min_distancia > DataGridView4(0, i).Value Then
                    indice_solucion_iteracion = i
                    min_distancia = DataGridView4(0, i).Value
                End If
            Next
            indice_solucion_iteracion = indice_solucion_iteracion
            'Colocando este codigo genetico en la tabla de P1
            For i = 0 To DataGridView5.ColumnCount - 1
                DataGridView5(i, 0).Value = DataGridView3(i,
indice_solucion_iteracion).Value
            Next
            'Buscando al segundo padre
            Dim indice_padre2 As Integer
            min_distancia = 1000000
            For i = 0 To DataGridView4.RowCount - 1
                If i <> indice_solucion_iteracion Then
                    If min_distancia > DataGridView4(0, i).Value Then
                        indice_padre2 = i
                        min_distancia = DataGridView4(0, i).Value
                    End If
                End If
            Next
            indice_padre2 = indice_padre2
            For i = 0 To DataGridView6.ColumnCount - 1
                DataGridView6(i, 0).Value = DataGridView3(i,
indice_padre2).Value
            Next
            '--------------------------------------
```

```
            ' el cruce ordenado por dos puntos
            '-------------------------------------
            Randomize()
            Dim corte1 As Integer
            Dim corte2 As Integer
            'CInt(Int((6 * Rnd()) + 1)) --> entre 1 y 6
            Dim aleatorio_1 As Integer
            aleatorio_1 = CInt(Int(((DataGridView5.ColumnCount - 1) * Rnd())
+ 0))
            Dim aleatorio_2 As Integer
            Dim po = 1
            While po = 1
                aleatorio_2 = CInt(Int(((DataGridView5.ColumnCount - 1) *
Rnd()) + 0))
                If aleatorio_2 <> aleatorio_1 Then
                    Exit While
                End If
            End While
            aleatorio_2 = aleatorio_2
            'Verificar el orden de los cortes
            If aleatorio_1 < aleatorio_2 Then
                corte1 = aleatorio_1
                corte2 = aleatorio_2
            ElseIf aleatorio_2 < aleatorio_1 Then
                corte1 = aleatorio_2
                corte2 = aleatorio_1
            End If
            corte1 = corte1
            corte2 = corte2
            TextBox1.Text = corte1
            TextBox2.Text = corte2
            '-------------------------
            '------ Cruce de genes
            'Preparando tabla de apoyo
            For i = 0 To DataGridView14.ColumnCount - 1
                DataGridView14(i, 0).Value = 0
            Next
            ''''''''''''''''''''P1 a P2 para H1
            For i = 0 To DataGridView12.ColumnCount - 1
                DataGridView12(i, 0).Value = -2
            Next
            For i = corte1 To corte2 'recorrido de cortes por el P1
                For j = 0 To DataGridView6.ColumnCount - 1 'recorrido por el
P2
                    If DataGridView5(i, 0).Value = DataGridView6(j, 0).Value
Then
                        DataGridView14(j, 0).Value = -1
                        Exit For
                    End If
                Next
            Next
            'Completando de P2 a H1
            Dim indice As Integer
            For i = corte1 To corte2
                For j = 0 To DataGridView14.ColumnCount - 1
                    If DataGridView14(j, 0).Value = -1 Then
                        indice = j
                        DataGridView14(indice, 0).Value = 0
                        Exit For
                    End If
                Next
                indice = indice
                DataGridView12(i, 0).Value = DataGridView6(indice, 0).Value
            Next
```

```
            'Completando de P1 a H1
            Dim resp As Integer
            For i = 0 To DataGridView5.ColumnCount - 1
                resp = 0
                For j = 0 To DataGridView12.ColumnCount - 1
                    If DataGridView5(i, 0).Value = DataGridView12(j, 0).Value
Then
                        resp = 1
                        Exit For
                    End If
                Next
                If resp = 0 Then
                    For j = 0 To DataGridView12.ColumnCount - 1
                        If DataGridView12(j, 0).Value = -2 Then
                            DataGridView12(j, 0).Value = DataGridView5(i,
0).Value
                            Exit For
                        End If
                    Next
                End If
            Next
            ''''''''''''''''''''''''''''''''''P2 a P1 para H2
            For i = 0 To DataGridView11.ColumnCount - 1
                DataGridView11(i, 0).Value = -2
            Next
            For i = corte1 To corte2 'recorrido de cortes por el P2
                For j = 0 To DataGridView5.ColumnCount - 1 'recorrido por el
P2
                    If DataGridView6(i, 0).Value = DataGridView5(j, 0).Value
Then
                        DataGridView14(j, 0).Value = -1
                        Exit For
                    End If
                Next
            Next
            'Completando de P1 a H2
            indice = 0
            For i = corte1 To corte2
                For j = 0 To DataGridView14.ColumnCount - 1
                    If DataGridView14(j, 0).Value = -1 Then
                        indice = j
                        DataGridView14(indice, 0).Value = 0
                        Exit For
                    End If
                Next
                indice = indice
                DataGridView11(i, 0).Value = DataGridView5(indice, 0).Value
            Next
            'Completando de P2 a H2
            resp = 0
            For i = 0 To DataGridView6.ColumnCount - 1
                resp = 0
                For j = 0 To DataGridView11.ColumnCount - 1
                    If DataGridView6(i, 0).Value = DataGridView11(j, 0).Value
Then
                        resp = 1
                        Exit For
                    End If
                Next
                If resp = 0 Then
                    For j = 0 To DataGridView11.ColumnCount - 1
                        If DataGridView11(j, 0).Value = -2 Then
                            DataGridView11(j, 0).Value = DataGridView6(i,
0).Value
```

```
                        Exit For
                    End If
                Next
            End If
        Next
        '-------------------------
        '------ Mutacion
        'Probabilidad de mutacion para H1
        Randomize()
        Dim mutacion As Double
        Dim indi_posi_1 As Integer
        Dim indi_posi_2 As Integer
        mutacion = Math.Round((1 * Rnd()) + 0, 4)
        If mutacion < 0.1 Then
            aleatorio_1 = CInt(Int(((DataGridView5.ColumnCount - 1) *
Rnd()) + 1)) 'No se considera punto inicial
            po = 1
            While po = 1
                aleatorio_2 = CInt(Int(((DataGridView5.ColumnCount - 1) *
Rnd()) + 1))
                If aleatorio_2 <> aleatorio_1 Then
                    Exit While
                End If
            End While
            aleatorio_2 = aleatorio_2
            'Verificar el orden de los cortes
            If aleatorio_1 < aleatorio_2 Then
                indi_posi_1 = aleatorio_1
                indi_posi_2 = aleatorio_2
            ElseIf aleatorio_2 < aleatorio_1 Then
                indi_posi_1 = aleatorio_2
                indi_posi_2 = aleatorio_1
            End If
            indi_posi_1 = indi_posi_1
            indi_posi_2 = indi_posi_2
            'Intercambiando posiciones
            Dim val1, val2 As Integer
            val1 = DataGridView12(indi_posi_1, 0).Value
            val2 = DataGridView12(indi_posi_2, 0).Value
            DataGridView12(indi_posi_2, 0).Value = val1
            DataGridView12(indi_posi_1, 0).Value = val2
        End If
        'Probabilidad de mutacion para H2
        Randomize()
        mutacion = Math.Round((1 * Rnd()) + 0, 4)
        If mutacion < 0.1 Then
            aleatorio_1 = CInt(Int(((DataGridView5.ColumnCount - 1) *
Rnd()) + 1)) 'No se considera punto inicial
            po = 1
            While po = 1
                aleatorio_2 = CInt(Int(((DataGridView5.ColumnCount - 1) *
Rnd()) + 1))
                If aleatorio_2 <> aleatorio_1 Then
                    Exit While
                End If
            End While
            aleatorio_2 = aleatorio_2
            'Verificar el orden de los cortes
            If aleatorio_1 < aleatorio_2 Then
                indi_posi_1 = aleatorio_1
                indi_posi_2 = aleatorio_2
            ElseIf aleatorio_2 < aleatorio_1 Then
                indi_posi_1 = aleatorio_2
                indi_posi_2 = aleatorio_1
```

```
                End If
                indi_posi_1 = indi_posi_1
                indi_posi_2 = indi_posi_2
                'Intercambiando posiciones
                Dim val1, val2 As Integer
                val1 = DataGridView11(indi_posi_1, 0).Value
                val2 = DataGridView11(indi_posi_2, 0).Value
                DataGridView11(indi_posi_2, 0).Value = val1
                DataGridView11(indi_posi_1, 0).Value = val2
            End If
            ''''''
```

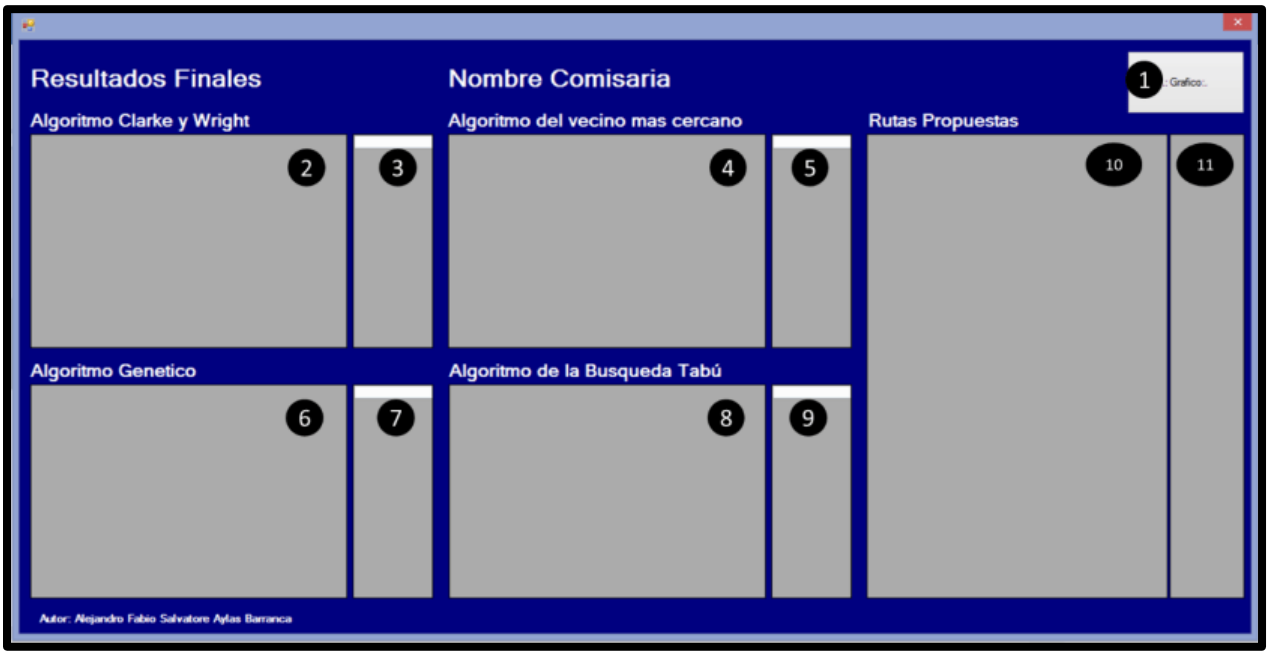
```
            min_distancia = 1000000
            For i = 0 To DataGridView4.RowCount - 1
                If min_distancia > DataGridView4(0, i).Value Then
                    indice_solucion_iteracion = i
                    min_distancia = DataGridView4(0, i).Value
                End If
            Next
            indice_solucion_iteracion = indice_solucion_iteracion
            DataGridView20.Rows.Add()
            DataGridView21.Rows.Add()
            DataGridView22.Rows.Add()
            DataGridView20(0, DataGridView20.RowCount - 1).Value = iteracion
            For i = 0 To DataGridView1.ColumnCount - 1
                DataGridView21(i, DataGridView21.RowCount - 1).Value =
DataGridView3(i, indice_solucion_iteracion).Value
            Next
            DataGridView22(0, DataGridView22.RowCount - 1).Value =
DataGridView4(0, indice_solucion_iteracion).Value
            'cambiando el codigo genetico por las etiquetas de los destinos
            For i = 0 To DataGridView21.ColumnCount - 1
                For t = 0 To DataGridView8.RowCount - 1
                    If DataGridView21(i, DataGridView21.RowCount - 1).Value =
DataGridView8(1, t).Value Then
                        DataGridView21(i, DataGridView21.RowCount - 1).Value
= DataGridView8(0, t).Value
                        Exit For
                    End If
                Next
            Next
            ' ------------------------------------------------------------
----
            ' -------   Dejando todo limpio para una nueva iteracion --------
----
            TextBox1.Text = 0
            TextBox2.Text = 0
            For i = 0 To DataGridView5.ColumnCount - 1
                DataGridView5(i, 0).Value = 0
                DataGridView6(i, 0).Value = 0
                DataGridView11(i, 0).Value = 0
                DataGridView12(i, 0).Value = 0
            Next
        Next
    End Sub
End Class
```

Anexo 10.8. Formulario SolucionesAG



Este formulario es llamado *SolucionesAG* los controles presentes son: en (1) está un DataGridView llamado *DataGridView1*, en (2) un DataGridView llamado *DataGridView2*

Anexo 10.9. Formulario ResultadosFinales



Este formulario es llamado *ResultadosFinales* los controles presentes son: en (1) está un Boton llamado *Button1*, en (2) un DataGridView llamado *DataGridView1,* en (3) está un DataGridView llamado *DataGridView2,* en (4) se encuentra un DataGridView llamado *DataGridView7,* en (5) se encuentra un DataGridView llamado *DataGridView8,* en (6) se encuentra un DataGridView llamado *DataGridView5,* en (7) se encuentra un DataGridView llamado *DataGridView6,* en (8) se encuentra un DataGridView llamado

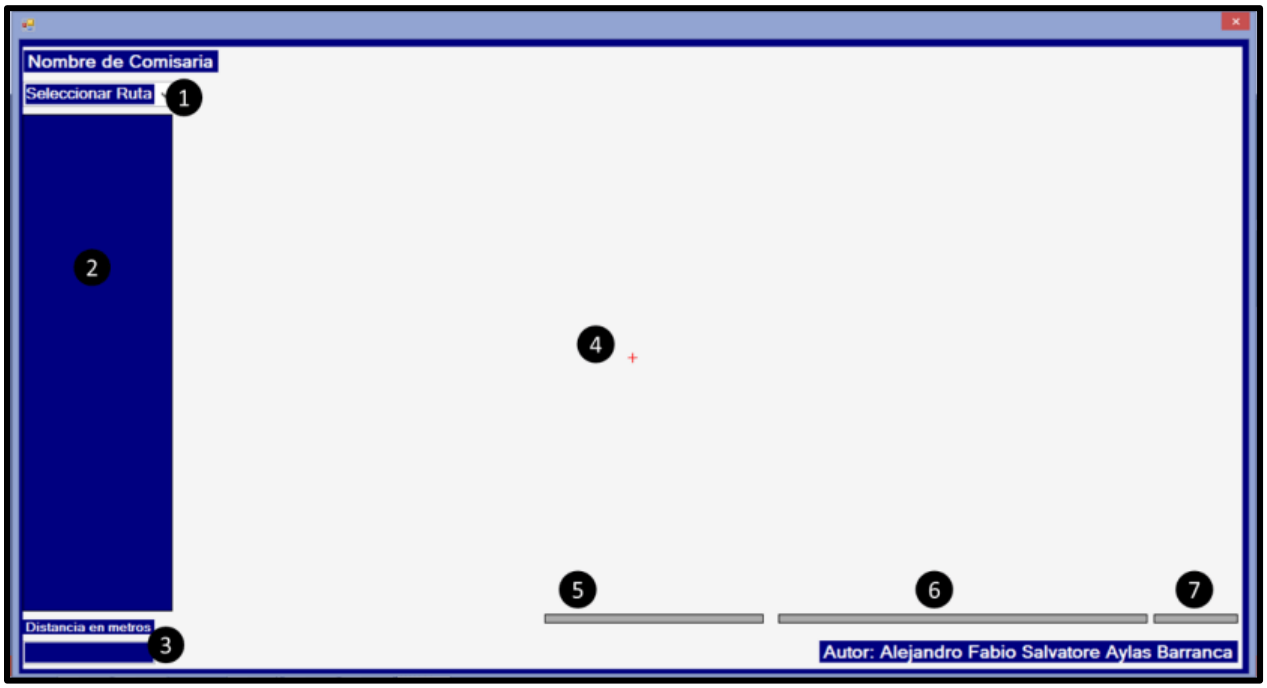
DataGridView3, en (9) se encuentra un DataGridView llamado *DataGridView4,* en (10) se encuentra un DataGridView llamado *DataGridView9* y en (11) está un DataGridView llamado *DataGridView10.* Los códigos de programación son:

```
Public Class ResultadosFinales
    Dim capacidad As Integer = Comisaria.NumPuntos.Text '> numero maximo de
destinos para visitar por ruta
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        MostrarRutas.Label1.Text = Label8.Text
        'Mandando rutas propuestas
        MostrarRutas.DataGridView2.ColumnCount = DataGridView9.ColumnCount
        MostrarRutas.DataGridView2.RowCount = DataGridView9.RowCount
        For i = 0 To DataGridView9.RowCount - 1
            For j = 0 To DataGridView9.ColumnCount - 1
                MostrarRutas.DataGridView2(j, i).Value = DataGridView9(j,
i).Value
            Next
        Next
        'las distancias
        MostrarRutas.DataGridView3.ColumnCount = DataGridView10.ColumnCount
        MostrarRutas.DataGridView3.RowCount = DataGridView10.RowCount
        For i = 0 To DataGridView10.RowCount - 1
            For j = 0 To DataGridView10.ColumnCount - 1
                MostrarRutas.DataGridView3(j, i).Value = DataGridView10(j,
i).Value
            Next
        Next
        'contar el numero de rutas obtenidos
        Dim num_rutas As Integer = DataGridView10.RowCount
        'Agregar items al combobox
        For i = 1 To num_rutas
            MostrarRutas.ComboBox1.Items.Add("Ruta " + i.ToString)
        Next
        MostrarRutas.DataGridView2.Visible = False
        MostrarRutas.DataGridView3.Visible = False
        MostrarRutas.DataGridView4.Visible = False
        MostrarRutas.Show()
    End Sub
    Sub Verificando_ruta_repetida()
        For i = 0 To DataGridView9.RowCount - 2
            Dim contador As Integer = 0
            For j = 0 To DataGridView9.ColumnCount - 1
                If (DataGridView9(j, DataGridView9.RowCount -
1).Value).ToString = (DataGridView9(j, i).Value).ToString Then
                    contador = contador + 1
                End If
            Next
            contador = contador
            If contador = DataGridView9.ColumnCount Then 'si coincide todos
los elementos
                'cortando la ultima fila del d9 y d10
                DataGridView9.RowCount = DataGridView9.RowCount - 1
                DataGridView10.RowCount = DataGridView10.RowCount - 1
                Exit For
            End If
        Next
    End Sub
    Private Sub ResultadosFinales_Load(sender As Object, e As EventArgs)
Handles MyBase.Load
        AlgoritmoGenetico.Close()
        'Filtrando las rutas
```

```
        DataGridView9.ColumnCount = capacidad + 2
        'Colocando las soluciones del metodo del ahorro
        For i = 0 To DataGridView1.RowCount - 1
            'Creando una nueva fila en la tabla de rutas filtradas
            DataGridView9.Rows.Add()
            For a = 0 To DataGridView9.ColumnCount - 1
                DataGridView9(a, DataGridView9.RowCount - 1).Value = 0
            Next
            DataGridView10.Rows.Add()
            'Pasando la ruta encontrada y su distancia correspondiente
            For a = 0 To DataGridView1.ColumnCount - 1
                DataGridView9(a, DataGridView9.RowCount - 1).Value =
DataGridView1(a, i).Value
            Next
            DataGridView10(0, DataGridView10.RowCount - 1).Value =
DataGridView2(0, i).Value
        Next
        'Colocando las soluciones del metodo vecino mas cercano
        For i = 0 To DataGridView7.RowCount - 1
            'Creando una nueva fila en la tabla de rutas filtradas
            DataGridView9.Rows.Add()
            For a = 0 To DataGridView9.ColumnCount - 1
                DataGridView9(a, DataGridView9.RowCount - 1).Value = 0
            Next
            DataGridView10.Rows.Add()
            'Pasando la ruta encontrada y su distancia correspondiente
            For a = 0 To DataGridView7.ColumnCount - 1
                DataGridView9(a, DataGridView9.RowCount - 1).Value =
DataGridView7(a, i).Value
            Next
            DataGridView10(0, DataGridView10.RowCount - 1).Value =
DataGridView8(0, i).Value
            'Verificando si la ruta añadida en la iteracion es identica a las
que ya estaban añadidas
            Verificando_ruta_repetida()
        Next
        'Colocando las soluciones del metodo tabu
        For i = 0 To DataGridView3.RowCount - 1
            'Creando una nueva fila en la tabla de rutas filtradas
            DataGridView9.Rows.Add()
            For a = 0 To DataGridView9.ColumnCount - 1
                DataGridView9(a, DataGridView9.RowCount - 1).Value = 0
            Next
            DataGridView10.Rows.Add()
            'Pasando la ruta encontrada y su distancia correspondiente
            For a = 0 To DataGridView3.ColumnCount - 1
                DataGridView9(a, DataGridView9.RowCount - 1).Value =
DataGridView3(a, i).Value
            Next
            DataGridView10(0, DataGridView10.RowCount - 1).Value =
DataGridView4(0, i).Value
            'Verificando si la ruta añadida en la iteracion es identica a las
que ya estaban añadidas
            Verificando_ruta_repetida()
        Next
        'Colocando las soluciones del Genetico
        For i = 0 To DataGridView5.RowCount - 1
            'Creando una nueva fila en la tabla de rutas filtradas
            DataGridView9.Rows.Add()
            For a = 0 To DataGridView9.ColumnCount - 1
                DataGridView9(a, DataGridView9.RowCount - 1).Value = 0
            Next
            DataGridView10.Rows.Add()
            'Pasando la ruta encontrada y su distancia correspondiente
```

```
            For a = 0 To DataGridView5.ColumnCount - 1
                DataGridView9(a, DataGridView9.RowCount - 1).Value =
DataGridView5(a, i).Value
            Next
            DataGridView10(0, DataGridView10.RowCount - 1).Value =
DataGridView6(0, i).Value
            'Verificando si la ruta añadida en la iteracion es identica a las
que ya estaban añadidas
            Verificando_ruta_repetida()
        Next
        'Buscando rutas con menor a cuatro destinos
        For i = 0 To DataGridView9.RowCount - 1
            Dim contador As Integer = 0
            If DataGridView9(5, i).Value = 0.ToString Then
                DataGridView10(0, i).Value = 0
            End If
            If DataGridView9(4, i).Value = 0.ToString Then
                DataGridView10(0, i).Value = 0
            End If
            If DataGridView9(3, i).Value = 0.ToString Then
                DataGridView10(0, i).Value = 0
            End If
        Next
        Dim a1, b1
        Dim c1, d1 As Integer
        'Ordenando de mayor a menor
        For i = 0 To DataGridView10.RowCount - 1
            For j = i + 1 To DataGridView10.RowCount - 1
                If DataGridView10(0, i).Value <= DataGridView10(0, j).Value
Then
                    For f = 0 To DataGridView9.ColumnCount - 1
                        a1 = DataGridView9(f, i).Value
                        b1 = DataGridView9(f, j).Value
                        DataGridView9(f, i).Value = b1
                        DataGridView9(f, j).Value = a1
                    Next
                    c1 = DataGridView10(0, i).Value
                    d1 = DataGridView10(0, j).Value
                    DataGridView10(0, i).Value = d1
                    DataGridView10(0, j).Value = c1
                End If
            Next
        Next
        'Cortando soluciones vacias
        For i = 0 To DataGridView10.RowCount - 1
            If DataGridView10(0, i).Value = 0.ToString Then
                DataGridView9.RowCount = i
                DataGridView10.RowCount = i
                Exit For
            End If
        Next
    End Sub
End Class
```

Anexo 10.10. Formulario MostrarRutas



Este formulario es llamado *MostrarRutas* los controles presentes son: en (1) está un ComboBox llamado *ComBox1*, en (2) un DataGridView llamado *DataGridView5,* en (3) está un TextBox llamado *TextBox1,* en (4) se encuentra un GMapControl llamado *GMapControl1,* en (5) se encuentra un DataGridView llamado *DataGridView4,* en (6) se encuentra un DataGridView llamado *DataGridView2* y en (7) está un DataGridView llamado *DataGridView3.* Los códigos de programación son:

```
Imports GMap.NET
Imports GMap.NET.MapProviders
Imports GMap.NET.WindowsForms
Imports GMap.NET.WindowsForms.Markers
Public Class MostrarRutas
    Dim marker As GMarkerGoogle
    Dim markerOverlay As GMapOverlay
    Dim dt As DataTable
    Dim capacidad As Integer = Comisaria.NumPuntos.Text '> numero maximo de
destinos para visitar por ruta
    Private Sub MostrarRutas_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        Me.Location = Screen.PrimaryScreen.WorkingArea.Location
        Me.Size = Screen.PrimaryScreen.WorkingArea.Size
        GMapControl1.DragButton = MouseButtons.Left
        GMapControl1.CanDragMap = True
        GMapControl1.MapProvider = GMapProviders.GoogleMap
        Marcar_Puntos_Seleccionados()

    End Sub
    Sub Marcar_Puntos_Seleccionados()
        If Label1.Text = "PIEDRA LIZA" Then
            Poligono_PD()
```

```
            GMapControl1.Position = New PointLatLng(-12.03876, -77.0197)
        ElseIf Label1.Text = "RIMAC" Then
            Poligono_RI()
            GMapControl1.Position = New PointLatLng(-12.03867, -77.03238)
        ElseIf Label1.Text = "MANZANO" Then
            Poligono_MA()
            GMapControl1.Position = New PointLatLng(-12.03009, -77.03245)
        ElseIf Label1.Text = "CIUDAD Y CAMPO" Then
            Poligono_CC()
            GMapControl1.Position = New PointLatLng(-12.01921, -77.03346)
        ElseIf Label1.Text = "FLOR DE AMANCAES" Then
            Poligono_FA()
            GMapControl1.Position = New PointLatLng(-12.01076, -77.03321)
        End If
    End Sub
    Private Sub ComboBox1_SelectedIndexChanged(sender As Object, e As
EventArgs) Handles ComboBox1.SelectedIndexChanged
        GMapControl1.Overlays.Clear()
        Marcar_Puntos_Seleccionados()
        DataGridView5.ColumnCount = 1
        DataGridView5.RowCount = DataGridView2.ColumnCount
        Dim num_rutas As Integer = DataGridView2.RowCount
        'Identificando la ruta seleccionada
        For i = 1 To num_rutas
            If ComboBox1.Text = "Ruta " + i.ToString Then
                'ruta1 ---> fila 0 del datagrid2
                DataGridView5.ColumnCount = 4
                'Colocando la Ruta
                For indice = 0 To DataGridView2.ColumnCount - 1
                    DataGridView5(0, indice).Value = DataGridView2(indice, i
- 1).Value
                Next
                'Colocando la distancia asociada a la ruta
                TextBox1.Text = DataGridView3(0, i - 1).Value
                Exit For
            End If
        Next
        'Ajustando la tabla de la Ruta
        For i = 0 To DataGridView5.RowCount - 1
            If DataGridView5(0, i).Value = "0".ToString Then
                DataGridView5.RowCount = i
                Exit For
            End If
        Next
        'Colocando las coordenadas Geograficas a cada Punto de la Ruta
Seleccionada
        Dim punto
        Dim coordenada_x
        Dim coordenada_y
        Dim descripcion
        If Label1.Text = "PIEDRA LIZA" Then
            For i = 0 To DataGridView5.RowCount - 1
                punto = DataGridView5(0, i).Value
                For j = 0 To CoordenadasGeograficas.DataGridView1.RowCount -
1
                    If punto = CoordenadasGeograficas.DataGridView1(0,
j).Value Then
                        coordenada_x =
CoordenadasGeograficas.DataGridView1(1, j).Value
                        coordenada_y =
CoordenadasGeograficas.DataGridView1(2, j).Value
                        descripcion = CoordenadasGeograficas.DataGridView1(3,
j).Value
                        DataGridView5(1, i).Value = coordenada_x
```

```
                        DataGridView5(2, i).Value = coordenada_y
                        DataGridView5(3, i).Value = descripcion
                        'Pintando de otro color los puntos cubiertos por ruta
                        ''''''Marcador
                        markerOverlay = New GMapOverlay("Marcador")
                        marker = New GMarkerGoogle(New
PointLatLng(coordenada_x, coordenada_y), GMarkerGoogleType.brown_small)
                        markerOverlay.Markers.Add(marker) 'Agregando el
marcador al GMapControl1
                        'Agregando un tooltipe al marcador generado
                        marker.ToolTipMode = MarkerTooltipMode.Always
                        marker.ToolTipText = String.Format("{0} ", punto)
                        'Agregando el GMapControl1 y el marcador al
mapcontrol
                        GMapControl1.Overlays.Add(markerOverlay)
                        Exit For
                    End If
                Next
            Next
        End If

        If Label1.Text = "RIMAC" Then
            For i = 0 To DataGridView5.RowCount - 1
                punto = DataGridView5(0, i).Value
                For j = 0 To CoordenadasGeograficas.DataGridView2.RowCount -
1
                    If punto = CoordenadasGeograficas.DataGridView2(0,
j).Value Then
                        coordenada_x =
CoordenadasGeograficas.DataGridView2(1, j).Value
                        coordenada_y =
CoordenadasGeograficas.DataGridView2(2, j).Value
                        descripcion = CoordenadasGeograficas.DataGridView2(3,
j).Value
                        DataGridView5(1, i).Value = coordenada_x
                        DataGridView5(2, i).Value = coordenada_y
                        DataGridView5(3, i).Value = descripcion
                        'Pintando de otro color los puntos cubiertos por ruta
                        ''''''Marcador
                        markerOverlay = New GMapOverlay("Marcador")
                        marker = New GMarkerGoogle(New
PointLatLng(coordenada_x, coordenada_y), GMarkerGoogleType.brown_small)
                        markerOverlay.Markers.Add(marker) 'Agregando el
marcador al GMapControl1
                        'Agregando un tooltipe al marcador generado
                        marker.ToolTipMode = MarkerTooltipMode.Always
                        marker.ToolTipText = String.Format("{0} ", punto)
                        'Agregando el GMapControl1 y el marcador al
mapcontrol
                        GMapControl1.Overlays.Add(markerOverlay)
                        Exit For
                    End If
                Next
            Next
        End If
        If Label1.Text = "MANZANO" Then
            For i = 0 To DataGridView5.RowCount - 1
                punto = DataGridView5(0, i).Value
                For j = 0 To CoordenadasGeograficas.DataGridView3.RowCount -
1
                    If punto = CoordenadasGeograficas.DataGridView3(0,
j).Value Then
                        coordenada_x =
CoordenadasGeograficas.DataGridView3(1, j).Value
```

```
                        coordenada_y =
CoordenadasGeograficas.DataGridView3(2, j).Value
                        descripcion = CoordenadasGeograficas.DataGridView3(3,
j).Value
                        DataGridView5(1, i).Value = coordenada_x
                        DataGridView5(2, i).Value = coordenada_y
                        DataGridView5(3, i).Value = descripcion
                        'Pintando de otro color los puntos cubiertos por ruta
                        ''''''Marcador
                        markerOverlay = New GMapOverlay("Marcador")
                        marker = New GMarkerGoogle(New
PointLatLng(coordenada_x, coordenada_y), GMarkerGoogleType.brown_small)
                        markerOverlay.Markers.Add(marker) 'Agregando el
marcador al GMapControl1
                        'Agregando un tooltipe al marcador generado
                        marker.ToolTipMode = MarkerTooltipMode.Always
                        marker.ToolTipText = String.Format("{0} ", punto)
                        'Agregando el GMapControl1 y el marcador al
mapcontrol
                        GMapControl1.Overlays.Add(markerOverlay)
                        Exit For
                    End If
                Next
            Next
        End If
        If Label1.Text = "CIUDAD Y CAMPO" Then
            For i = 0 To DataGridView5.RowCount - 1
                punto = DataGridView5(0, i).Value
                For j = 0 To CoordenadasGeograficas.DataGridView4.RowCount -
1
                    If punto = CoordenadasGeograficas.DataGridView4(0,
j).Value Then
                        coordenada_x =
CoordenadasGeograficas.DataGridView4(1, j).Value
                        coordenada_y =
CoordenadasGeograficas.DataGridView4(2, j).Value
                        descripcion = CoordenadasGeograficas.DataGridView4(3,
j).Value
                        DataGridView5(1, i).Value = coordenada_x
                        DataGridView5(2, i).Value = coordenada_y
                        DataGridView5(3, i).Value = descripcion
                        'Pintando de otro color los puntos cubiertos por ruta
                        ''''''Marcador
                        markerOverlay = New GMapOverlay("Marcador")
                        marker = New GMarkerGoogle(New
PointLatLng(coordenada_x, coordenada_y), GMarkerGoogleType.brown_small)
                        markerOverlay.Markers.Add(marker) 'Agregando el
marcador al GMapControl1
                        'Agregando un tooltipe al marcador generado
                        marker.ToolTipMode = MarkerTooltipMode.Always
                        marker.ToolTipText = String.Format("{0} ", punto)
                        'Agregando el GMapControl1 y el marcador al
mapcontrol
                        GMapControl1.Overlays.Add(markerOverlay)
                        Exit For
                    End If
                Next
            Next
        End If
        If Label1.Text = "FLOR DE AMANCAES" Then
            For i = 0 To DataGridView5.RowCount - 1
                punto = DataGridView5(0, i).Value
                For j = 0 To CoordenadasGeograficas.DataGridView5.RowCount -
1
```

```
                    If punto = CoordenadasGeograficas.DataGridView5(0,
j).Value Then
                        coordenada_x =
CoordenadasGeograficas.DataGridView5(1, j).Value
                        coordenada_y =
CoordenadasGeograficas.DataGridView5(2, j).Value
                        descripcion = CoordenadasGeograficas.DataGridView5(3,
j).Value
                        DataGridView5(1, i).Value = coordenada_x
                        DataGridView5(2, i).Value = coordenada_y
                        DataGridView5(3, i).Value = descripcion
                        'Pintando de otro color los puntos cubiertos por ruta
                        ''''''Marcador
                        markerOverlay = New GMapOverlay("Marcador")
                        marker = New GMarkerGoogle(New
PointLatLng(coordenada_x, coordenada_y), GMarkerGoogleType.brown_small)
                        markerOverlay.Markers.Add(marker) 'Agregando el
marcador al GMapControl1
                        'Agregando un tooltipe al marcador generado
                        marker.ToolTipMode = MarkerTooltipMode.Always
                        marker.ToolTipText = String.Format("{0} ", punto)
                        'Agregando el GMapControl1 y el marcador al
mapcontrol
                        GMapControl1.Overlays.Add(markerOverlay)
                        Exit For
                    End If
                Next
            Next
        End If
        'Trazando la ruta
        'Ocultando las coordenadas geograficas del datagrid1
        DataGridView5.Columns(1).Visible = False
        DataGridView5.Columns(2).Visible = False
        Dim Ruta As GMapOverlay = New GMapOverlay("CapaRuta")
        Dim puntos As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        Dim lat As Double
        Dim lng As Double
        For i = 0 To DataGridView5.RowCount - 1
            lat = Convert.ToDouble(DataGridView5.Rows(i).Cells(1).Value)
            lng = Convert.ToDouble(DataGridView5.Rows(i).Cells(2).Value)
            puntos.Add(New PointLatLng(lat, lng))
        Next
        Dim PuntosRuta As GMapRoute = New GMapRoute(puntos, "Ruta")
        PuntosRuta.Stroke = New Pen(Color.Red, 3)
        Ruta.Routes.Add(PuntosRuta)
        GMapControl1.Overlays.Add(Ruta)
        'Actualizar el GMapControl1
        GMapControl1.Zoom = GMapControl1.Zoom + 1
        GMapControl1.Zoom = GMapControl1.Zoom - 1
        ''Ordenando la tabla de Ruta
        'Pasando toda la informacion a tabla de apoyo
        DataGridView4.ColumnCount = DataGridView5.ColumnCount
        DataGridView4.RowCount = DataGridView5.RowCount
        For i = 0 To DataGridView4.RowCount - 1
            For j = 0 To DataGridView4.ColumnCount - 1
                DataGridView4(j, i).Value = DataGridView5(j, i).Value
                DataGridView5(j, i).Value = ""
            Next
        Next
        DataGridView5.RowCount = DataGridView5.RowCount +
DataGridView5.RowCount - 1
        For i = 1 To DataGridView5.RowCount - 1 Step 2
            DataGridView5(0, i).Value = "   " "
```

```
        Next
        'Completando con la informacion
        For i = 0 To DataGridView4.RowCount - 1
            DataGridView5(0, i * 2).Value = DataGridView4(0, i).Value
            DataGridView5(1, i * 2).Value = DataGridView4(1, i).Value
            DataGridView5(2, i * 2).Value = DataGridView4(2, i).Value
            DataGridView5(3, i * 2).Value = DataGridView4(3, i).Value
        Next
        DataGridView5.Columns(0).Width = 40
        DataGridView5.Columns(1).Width = 100
    End Sub
    Sub Poligono_MA()
        GMapControl1.DragButton = MouseButtons.Left
        GMapControl1.CanDragMap = True
        GMapControl1.MapProvider = GMapProviders.GoogleMap
        GMapControl1.Position = New PointLatLng(-12.03009, -77.03245)
        GMapControl1.MinZoom = 0
        GMapControl1.MaxZoom = 24
        GMapControl1.Zoom = 15
        GMapControl1.AutoScroll = True
        'Agregando los poligonos del area de jurisdiccion
        Dim S1 As DataTable
        S1 = New DataTable()
        S1.Columns.Add("")
        S1.Columns.Add("")
        S1.Rows.Add(-12.03867, -77.03232)
        S1.Rows.Add(-12.03371, -77.03879)
        S1.Rows.Add(-12.0302, -77.0359)
        S1.Rows.Add(-12.03494, -77.02979)
        S1.Rows.Add(-12.03867, -77.03232)
        Dim Poligono_S1 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        Dim lat As Double
        Dim lng As Double
        For i = 0 To S1.Rows.Count - 1
            lat = Convert.ToDouble(S1(i)(0))
            lng = Convert.ToDouble(S1(i)(1))
            puntos.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto As GMapPolygon = New GMapPolygon(puntos,
"Poligono")
        Poligono_S1.Polygons.Add(poligonosPunto)
        GMapControl1.Overlays.Add(Poligono_S1)
        Dim S2 As DataTable
        S2 = New DataTable()
        S2.Columns.Add("")
        S2.Columns.Add("")
        S2.Rows.Add(-12.03371, -77.03879)
        S2.Rows.Add(-12.02988, -77.04287)
        S2.Rows.Add(-12.02673, -77.04)
        S2.Rows.Add(-12.03005, -77.03582)
        S2.Rows.Add(-12.0302, -77.0359)
        S2.Rows.Add(-12.03371, -77.03879)
        Dim Poligono_S2 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos2 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S2.Rows.Count - 1
            lat = Convert.ToDouble(S2(i)(0))
            lng = Convert.ToDouble(S2(i)(1))
            puntos2.Add(New PointLatLng(lat, lng))
        Next
```

```
        Dim poligonosPunto2 As GMapPolygon = New GMapPolygon(puntos2,
"Poligono")
        Poligono_S2.Polygons.Add(poligonosPunto2)
        GMapControl1.Overlays.Add(Poligono_S2)
        Dim S3 As DataTable
        S3 = New DataTable()
        S3.Columns.Add("")
        S3.Columns.Add("")
        S3.Rows.Add(-12.03494, -77.02979)
        S3.Rows.Add(-12.0302, -77.0359)
        S3.Rows.Add(-12.03005, -77.03582)
        S3.Rows.Add(-12.02695, -77.03331)
        S3.Rows.Add(-12.02729, -77.03292)
        S3.Rows.Add(-12.03178, -77.02771)
        S3.Rows.Add(-12.03494, -77.02979)
        Dim Poligono_S3 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos3 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S3.Rows.Count - 1
            lat = Convert.ToDouble(S3(i)(0))
            lng = Convert.ToDouble(S3(i)(1))
            puntos3.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto3 As GMapPolygon = New GMapPolygon(puntos3,
"Poligono")
        Poligono_S3.Polygons.Add(poligonosPunto3)
        GMapControl1.Overlays.Add(Poligono_S3)
        Dim S4 As DataTable
        S4 = New DataTable()
        S4.Columns.Add("")
        S4.Columns.Add("")
        S4.Rows.Add(-12.03005, -77.03582)
        S4.Rows.Add(-12.02673, -77.04)
        S4.Rows.Add(-12.02361, -77.03714)
        S4.Rows.Add(-12.02695, -77.03331)
        S4.Rows.Add(-12.03005, -77.03582)
        Dim Poligono_S4 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos4 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S4.Rows.Count - 1
            lat = Convert.ToDouble(S4(i)(0))
            lng = Convert.ToDouble(S4(i)(1))
            puntos4.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto4 As GMapPolygon = New GMapPolygon(puntos4,
"Poligono")
        Poligono_S4.Polygons.Add(poligonosPunto4)
        GMapControl1.Overlays.Add(Poligono_S4)
        Dim S5 As DataTable
        S5 = New DataTable()
        S5.Columns.Add("")
        S5.Columns.Add("")
        S5.Rows.Add(-12.03178, -77.02771)
        S5.Rows.Add(-12.02729, -77.03292)
        S5.Rows.Add(-12.0245, -77.03032)
        S5.Rows.Add(-12.02736, -77.02741)
        S5.Rows.Add(-12.02766, -77.02742)
        S5.Rows.Add(-12.02797, -77.02754)
        S5.Rows.Add(-12.02777, -77.02734)
        S5.Rows.Add(-12.02768, -77.02698)
        S5.Rows.Add(-12.0277, -77.02668)
        S5.Rows.Add(-12.02946, -77.02445)
```

```
        S5.Rows.Add(-12.03178, -77.02771)
        Dim Poligono_S5 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos5 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S5.Rows.Count - 1
            lat = Convert.ToDouble(S5(i)(0))
            lng = Convert.ToDouble(S5(i)(1))
            puntos5.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto5 As GMapPolygon = New GMapPolygon(puntos5,
"Poligono")
        Poligono_S5.Polygons.Add(poligonosPunto5)
        GMapControl1.Overlays.Add(Poligono_S5)
        Dim S6 As DataTable
        S6 = New DataTable()
        S6.Columns.Add("")
        S6.Columns.Add("")
        S6.Rows.Add(-12.02729, -77.03292)
        S6.Rows.Add(-12.02695, -77.03331)
        S6.Rows.Add(-12.02361, -77.03714)
        S6.Rows.Add(-12.02238, -77.03607)
        S6.Rows.Add(-12.02293, -77.03543)
        S6.Rows.Add(-12.02281, -77.03343)
        S6.Rows.Add(-12.02173, -77.03244)
        S6.Rows.Add(-12.02282, -77.03103)
        S6.Rows.Add(-12.02333, -77.03113)
        S6.Rows.Add(-12.02385, -77.03101)
        S6.Rows.Add(-12.0245, -77.03032)
        S6.Rows.Add(-12.02729, -77.03292)
        Dim Poligono_S6 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos6 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S6.Rows.Count - 1
            lat = Convert.ToDouble(S6(i)(0))
            lng = Convert.ToDouble(S6(i)(1))
            puntos6.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto6 As GMapPolygon = New GMapPolygon(puntos6,
"Poligono")
        Poligono_S6.Polygons.Add(poligonosPunto6)
        GMapControl1.Overlays.Add(Poligono_S6)
        GMapControl1.Zoom = GMapControl1.Zoom + 1
        GMapControl1.Zoom = GMapControl1.Zoom - 1
    End Sub
    Sub Poligono_RI()
        GMapControl1.DragButton = MouseButtons.Left
        GMapControl1.CanDragMap = True
        GMapControl1.MapProvider = GMapProviders.GoogleMap
        GMapControl1.Position = New PointLatLng(-12.03867, -77.03238)
        GMapControl1.MinZoom = 0
        GMapControl1.MaxZoom = 24
        GMapControl1.Zoom = 15
        GMapControl1.AutoScroll = True
        'Agregando los poligonos del area de jurisdiccion
        Dim S1 As DataTable
        S1 = New DataTable()
        S1.Columns.Add("")
        S1.Columns.Add("")
        S1.Rows.Add(-12.03732, -77.04058)
        S1.Rows.Add(-12.03667, -77.04345)
        S1.Rows.Add(-12.02829, -77.04419)
        S1.Rows.Add(-12.0298, -77.04284)
```

```
        S1.Rows.Add(-12.03065, -77.04215)
        S1.Rows.Add(-12.03221, -77.04038)
        S1.Rows.Add(-12.03444, -77.03793)
        S1.Rows.Add(-12.03605, -77.03937)
        S1.Rows.Add(-12.03732, -77.04058)
        Dim Poligono_S1 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        Dim lat As Double
        Dim lng As Double
        For i = 0 To S1.Rows.Count - 1
            lat = Convert.ToDouble(S1(i)(0))
            lng = Convert.ToDouble(S1(i)(1))
            puntos.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto As GMapPolygon = New GMapPolygon(puntos,
"Poligono")
        Poligono_S1.Polygons.Add(poligonosPunto)
        GMapControl1.Overlays.Add(Poligono_S1)
        Dim S2 As DataTable
        S2 = New DataTable()
        S2.Columns.Add("")
        S2.Columns.Add("")
        S2.Rows.Add(-12.03667, -77.04345)
        S2.Rows.Add(-12.03773, -77.04331)
        S2.Rows.Add(-12.03839, -77.0421)
        S2.Rows.Add(-12.03862, -77.04008)
        S2.Rows.Add(-12.03905, -77.0382)
        S2.Rows.Add(-12.04092, -77.03371)
        S2.Rows.Add(-12.0387, -77.03227)
        S2.Rows.Add(-12.03444, -77.03793)
        S2.Rows.Add(-12.03605, -77.03937)
        S2.Rows.Add(-12.03732, -77.04058)
        S2.Rows.Add(-12.03667, -77.04345)
        Dim Poligono_S2 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos2 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S2.Rows.Count - 1
            lat = Convert.ToDouble(S2(i)(0))
            lng = Convert.ToDouble(S2(i)(1))
            puntos2.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto2 As GMapPolygon = New GMapPolygon(puntos2,
"Poligono")
        Poligono_S2.Polygons.Add(poligonosPunto2)
        GMapControl1.Overlays.Add(Poligono_S2)
        Dim S3 As DataTable
        S3 = New DataTable()
        S3.Columns.Add("")
        S3.Columns.Add("")
        S3.Rows.Add(-12.04092, -77.03371)
        S3.Rows.Add(-12.03871, -77.03228)
        S3.Rows.Add(-12.03983, -77.0309)
        S3.Rows.Add(-12.0401, -77.02837)
        S3.Rows.Add(-12.04034, -77.02253)
        S3.Rows.Add(-12.04197, -77.02257)
        S3.Rows.Add(-12.04226, -77.02386)
        S3.Rows.Add(-12.04252, -77.02383)
        S3.Rows.Add(-12.04255, -77.02394)
        S3.Rows.Add(-12.04339, -77.02436)
        S3.Rows.Add(-12.04303, -77.02906)
        S3.Rows.Add(-12.04092, -77.03371)
```

```
        Dim Poligono_S3 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos3 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S3.Rows.Count - 1
            lat = Convert.ToDouble(S3(i)(0))
            lng = Convert.ToDouble(S3(i)(1))
            puntos3.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto3 As GMapPolygon = New GMapPolygon(puntos3,
"Poligono")
        Poligono_S3.Polygons.Add(poligonosPunto3)
        GMapControl1.Overlays.Add(Poligono_S3)
        Dim S4 As DataTable
        S4 = New DataTable()
        S4.Columns.Add("")
        S4.Columns.Add("")
        S4.Rows.Add(-12.0401, -77.02837)
        S4.Rows.Add(-12.03983, -77.0309)
        S4.Rows.Add(-12.03871, -77.03228)
        S4.Rows.Add(-12.03201, -77.02776)
        S4.Rows.Add(-12.03482, -77.02516)
        S4.Rows.Add(-12.03783, -77.02683)
        S4.Rows.Add(-12.03795, -77.0271)
        S4.Rows.Add(-12.0401, -77.02837)
        Dim Poligono_S4 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos4 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S4.Rows.Count - 1
            lat = Convert.ToDouble(S4(i)(0))
            lng = Convert.ToDouble(S4(i)(1))
            puntos4.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto4 As GMapPolygon = New GMapPolygon(puntos4,
"Poligono")
        Poligono_S4.Polygons.Add(poligonosPunto4)
        GMapControl1.Overlays.Add(Poligono_S4)

        Dim S5 As DataTable
        S5 = New DataTable()
        S5.Columns.Add("")
        S5.Columns.Add("")
        S5.Rows.Add(-12.04034, -77.02253)
        S5.Rows.Add(-12.03903, -77.02298)
        S5.Rows.Add(-12.03876, -77.02232)
        S5.Rows.Add(-12.03714, -77.02148)
        S5.Rows.Add(-12.03669, -77.02154)
        S5.Rows.Add(-12.03641, -77.02135)
        S5.Rows.Add(-12.03605, -77.02124)
        S5.Rows.Add(-12.03543, -77.02072)
        S5.Rows.Add(-12.03484, -77.02059)
        S5.Rows.Add(-12.03453, -77.02077)
        S5.Rows.Add(-12.03425, -77.02123)
        S5.Rows.Add(-12.03425, -77.02216)
        S5.Rows.Add(-12.03453, -77.02339)
        S5.Rows.Add(-12.03437, -77.02485)
        S5.Rows.Add(-12.03482, -77.02516)
        S5.Rows.Add(-12.03783, -77.02683)
        S5.Rows.Add(-12.03795, -77.0271)
        S5.Rows.Add(-12.0401, -77.02837)
        S5.Rows.Add(-12.04034, -77.02253)
        Dim Poligono_S5 As GMapOverlay = New GMapOverlay("Poligono")
```

```
        Dim puntos5 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S5.Rows.Count - 1
            lat = Convert.ToDouble(S5(i)(0))
            lng = Convert.ToDouble(S5(i)(1))
            puntos5.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto5 As GMapPolygon = New GMapPolygon(puntos5,
"Poligono")
        Poligono_S5.Polygons.Add(poligonosPunto5)
        GMapControl1.Overlays.Add(Poligono_S5)
        Dim S6 As DataTable
        S6 = New DataTable()
        S6.Columns.Add("")
        S6.Columns.Add("")
        S6.Rows.Add(-12.03482, -77.02516)
        S6.Rows.Add(-12.03201, -77.02776)
        S6.Rows.Add(-12.03192, -77.02768)
        S6.Rows.Add(-12.03184, -77.02778)
        S6.Rows.Add(-12.03179, -77.02771)
        S6.Rows.Add(-12.03119, -77.02803)
        S6.Rows.Add(-12.02906, -77.02534)
        S6.Rows.Add(-12.02939, -77.02458)
        S6.Rows.Add(-12.02923, -77.02388)
        S6.Rows.Add(-12.02955, -77.02307)
        S6.Rows.Add(-12.03036, -77.02237)
        S6.Rows.Add(-12.03186, -77.02176)
        S6.Rows.Add(-12.03232, -77.02133)
        S6.Rows.Add(-12.03365, -77.02086)
        S6.Rows.Add(-12.03484, -77.02059)
        S6.Rows.Add(-12.03453, -77.02077)
        S6.Rows.Add(-12.03425, -77.02123)
        S6.Rows.Add(-12.03425, -77.02216)
        S6.Rows.Add(-12.03453, -77.02339)
        S6.Rows.Add(-12.03437, -77.02485)
        S6.Rows.Add(-12.03482, -77.02516)
        Dim Poligono_S6 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos6 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S6.Rows.Count - 1
            lat = Convert.ToDouble(S6(i)(0))
            lng = Convert.ToDouble(S6(i)(1))
            puntos6.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto6 As GMapPolygon = New GMapPolygon(puntos6,
"Poligono")
        Poligono_S6.Polygons.Add(poligonosPunto6)
        GMapControl1.Overlays.Add(Poligono_S6)
        GMapControl1.Zoom = GMapControl1.Zoom + 1
        GMapControl1.Zoom = GMapControl1.Zoom - 1
    End Sub
    Sub Poligono_PD()
        GMapControl1.DragButton = MouseButtons.Left
        GMapControl1.CanDragMap = True
        GMapControl1.MapProvider = GMapProviders.GoogleMap
        GMapControl1.Position = New PointLatLng(-12.03876, -77.0197)
        GMapControl1.MinZoom = 0
        GMapControl1.MaxZoom = 24
        GMapControl1.Zoom = 15
        GMapControl1.AutoScroll = True
        'Agregando GMapControl1 con colores
        Dim S1_1 As DataTable
```

```
        S1_1 = New DataTable()
        S1_1.Columns.Add("")
        S1_1.Columns.Add("")
        S1_1.Rows.Add(-12.04187, -77.0226)
        S1_1.Rows.Add(-12.04151, -77.02255)
        S1_1.Rows.Add(-12.04113, -77.02255)
        S1_1.Rows.Add(-12.0404, -77.02257)
        S1_1.Rows.Add(-12.04035, -77.02275)
        S1_1.Rows.Add(-12.03902, -77.02292)
        S1_1.Rows.Add(-12.03897, -77.0226)
        S1_1.Rows.Add(-12.03869, -77.02228)
        S1_1.Rows.Add(-12.0381, -77.02195)
        S1_1.Rows.Add(-12.03759, -77.02183)
        S1_1.Rows.Add(-12.03722, -77.02155)
        S1_1.Rows.Add(-12.03698, -77.02154)
        S1_1.Rows.Add(-12.0367, -77.02156)
        S1_1.Rows.Add(-12.03641, -77.02138)
        S1_1.Rows.Add(-12.03608, -77.02131)
        S1_1.Rows.Add(-12.03601, -77.02127)
        S1_1.Rows.Add(-12.0365, -77.0208)
        S1_1.Rows.Add(-12.0377, -77.02041)
        S1_1.Rows.Add(-12.03768, -77.01997)
        S1_1.Rows.Add(-12.03786, -77.01929)
        S1_1.Rows.Add(-12.03831, -77.01897)
        S1_1.Rows.Add(-12.03833, -77.01861)
        S1_1.Rows.Add(-12.03818, -77.01823)
        S1_1.Rows.Add(-12.03856, -77.01782)
        S1_1.Rows.Add(-12.0389, -77.01784)
        S1_1.Rows.Add(-12.03925, -77.01761)
        S1_1.Rows.Add(-12.03938, -77.01799)
        S1_1.Rows.Add(-12.04009, -77.01808)
        S1_1.Rows.Add(-12.04039, -77.01783)
        S1_1.Rows.Add(-12.04054, -77.01816)
        S1_1.Rows.Add(-12.04057, -77.01846)
        S1_1.Rows.Add(-12.04085, -77.01935)
        S1_1.Rows.Add(-12.04119, -77.02036)
        S1_1.Rows.Add(-12.04161, -77.02163)
        S1_1.Rows.Add(-12.04187, -77.0226)
        Dim Poligono_S1_1 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        Dim lat As Double
        Dim lng As Double
        For i = 0 To S1_1.Rows.Count - 1
            lat = Convert.ToDouble(S1_1(i)(0))
            lng = Convert.ToDouble(S1_1(i)(1))
            puntos.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto As GMapPolygon = New GMapPolygon(puntos,
"Poligono")
        Poligono_S1_1.Polygons.Add(poligonosPunto)
        GMapControl1.Overlays.Add(Poligono_S1_1)
        Dim S1_2 As DataTable
        S1_2 = New DataTable()
        S1_2.Columns.Add("")
        S1_2.Columns.Add("")
        S1_2.Rows.Add(-12.04039, -77.01783)
        S1_2.Rows.Add(-12.04009, -77.01808)
        S1_2.Rows.Add(-12.03938, -77.01799)
        S1_2.Rows.Add(-12.03925, -77.01761)
        S1_2.Rows.Add(-12.0389, -77.01784)
        S1_2.Rows.Add(-12.03856, -77.01782)
        S1_2.Rows.Add(-12.03818, -77.01823)
```

```
        S1_2.Rows.Add(-12.03833, -77.01861)
        S1_2.Rows.Add(-12.03831, -77.01897)
        S1_2.Rows.Add(-12.03786, -77.01929)
        S1_2.Rows.Add(-12.03768, -77.01997)
        S1_2.Rows.Add(-12.0377, -77.02041)
        S1_2.Rows.Add(-12.0365, -77.0208)
        S1_2.Rows.Add(-12.03601, -77.02127)
        S1_2.Rows.Add(-12.03561, -77.02087)
        S1_2.Rows.Add(-12.03527, -77.02065)
        S1_2.Rows.Add(-12.03496, -77.0206)
        S1_2.Rows.Add(-12.0347, -77.02068)
        S1_2.Rows.Add(-12.03459, -77.02077)
        S1_2.Rows.Add(-12.03441, -77.02092)
        S1_2.Rows.Add(-12.03435, -77.02114)
        S1_2.Rows.Add(-12.03421, -77.02123)
        S1_2.Rows.Add(-12.03382, -77.02062)
        S1_2.Rows.Add(-12.03354, -77.01999)
        S1_2.Rows.Add(-12.03333, -77.01987)
        S1_2.Rows.Add(-12.033, -77.01987)
        S1_2.Rows.Add(-12.03292, -77.01965)
        S1_2.Rows.Add(-12.03207, -77.01908)
        S1_2.Rows.Add(-12.03243, -77.01853)
        S1_2.Rows.Add(-12.03249, -77.01824)
        S1_2.Rows.Add(-12.03277, -77.01779)
        S1_2.Rows.Add(-12.03371, -77.01765)
        S1_2.Rows.Add(-12.03415, -77.0178)
        S1_2.Rows.Add(-12.03389, -77.01734)
        S1_2.Rows.Add(-12.03381, -77.01707)
        S1_2.Rows.Add(-12.03383, -77.01687)
        S1_2.Rows.Add(-12.03396, -77.01664)
        S1_2.Rows.Add(-12.03468, -77.01618)
        S1_2.Rows.Add(-12.03497, -77.01609)
        S1_2.Rows.Add(-12.03562, -77.0157)
        S1_2.Rows.Add(-12.03625, -77.01565)
        S1_2.Rows.Add(-12.03648, -77.01555)
        S1_2.Rows.Add(-12.03661, -77.01538)
        S1_2.Rows.Add(-12.03664, -77.01516)
        S1_2.Rows.Add(-12.03693, -77.01482)
        S1_2.Rows.Add(-12.0371, -77.01471)
        S1_2.Rows.Add(-12.03696, -77.01449)
        S1_2.Rows.Add(-12.03717, -77.01416)
        S1_2.Rows.Add(-12.03724, -77.01362)
        S1_2.Rows.Add(-12.03742, -77.01327)
        S1_2.Rows.Add(-12.03849, -77.01519)
        S1_2.Rows.Add(-12.03956, -77.01655)
        S1_2.Rows.Add(-12.04039, -77.01783)
        Dim Poligono_S1_2 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos2 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S1_2.Rows.Count - 1
            lat = Convert.ToDouble(S1_2(i)(0))
            lng = Convert.ToDouble(S1_2(i)(1))
            puntos2.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto2 As GMapPolygon = New GMapPolygon(puntos2,
"Poligono")
        Poligono_S1_2.Polygons.Add(poligonosPunto2)
        GMapControl1.Overlays.Add(Poligono_S1_2)
        'Actualizar el GMapControl1
        GMapControl1.Zoom = GMapControl1.Zoom + 1
        GMapControl1.Zoom = GMapControl1.Zoom - 1
        Dim S2_1 As DataTable
        S2_1 = New DataTable()
```

```
        S2_1.Columns.Add("")
        S2_1.Columns.Add("")
        S2_1.Rows.Add(-12.04187, -77.0226)
        S2_1.Rows.Add(-12.0422, -77.02388)
        S2_1.Rows.Add(-12.04252, -77.02384)
        S2_1.Rows.Add(-12.04277, -77.02401)
        S2_1.Rows.Add(-12.04346, -77.02435)
        S2_1.Rows.Add(-12.04352, -77.023)
        S2_1.Rows.Add(-12.04328, -77.02125)
        S2_1.Rows.Add(-12.04357, -77.01929)
        S2_1.Rows.Add(-12.0422, -77.01982)
        S2_1.Rows.Add(-12.04119, -77.02036)
        S2_1.Rows.Add(-12.04161, -77.02163)
        S2_1.Rows.Add(-12.04187, -77.0226)
        Dim Poligono_S2_1 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos3 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S2_1.Rows.Count - 1
            lat = Convert.ToDouble(S2_1(i)(0))
            lng = Convert.ToDouble(S2_1(i)(1))
            puntos3.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto3 As GMapPolygon = New GMapPolygon(puntos3,
"Poligono")
        Poligono_S2_1.Polygons.Add(poligonosPunto3)
        GMapControl1.Overlays.Add(Poligono_S2_1)
        Dim S2_2 As DataTable
        S2_2 = New DataTable()
        S2_2.Columns.Add("")
        S2_2.Columns.Add("")
        S2_2.Rows.Add(-12.04119, -77.02036)
        S2_2.Rows.Add(-12.04085, -77.01935)
        S2_2.Rows.Add(-12.04057, -77.01846)
        S2_2.Rows.Add(-12.04054, -77.01816)
        S2_2.Rows.Add(-12.04039, -77.01783)
        S2_2.Rows.Add(-12.03956, -77.01655)
        S2_2.Rows.Add(-12.03849, -77.01519)
        S2_2.Rows.Add(-12.03742, -77.01327)
        S2_2.Rows.Add(-12.03784, -77.01332)
        S2_2.Rows.Add(-12.03899, -77.01384)
        S2_2.Rows.Add(-12.04085, -77.01429)
        S2_2.Rows.Add(-12.04172, -77.01504)
        S2_2.Rows.Add(-12.04318, -77.01697)
        S2_2.Rows.Add(-12.04353, -77.01797)
        S2_2.Rows.Add(-12.04357, -77.01929)
        S2_2.Rows.Add(-12.0422, -77.01982)
        S2_2.Rows.Add(-12.04119, -77.02036)
        Dim Poligono_S2_2 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos4 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S2_2.Rows.Count - 1
            lat = Convert.ToDouble(S2_2(i)(0))
            lng = Convert.ToDouble(S2_2(i)(1))
            puntos4.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto4 As GMapPolygon = New GMapPolygon(puntos4,
"Poligono")
        Poligono_S2_2.Polygons.Add(poligonosPunto4)
        GMapControl1.Overlays.Add(Poligono_S2_2)
        GMapControl1.Zoom = GMapControl1.Zoom + 1
        GMapControl1.Zoom = GMapControl1.Zoom - 1
    End Sub
```

```
    Sub Poligono_CC()
        GMapControl1.DragButton = MouseButtons.Left
        GMapControl1.CanDragMap = True
        GMapControl1.MapProvider = GMapProviders.GoogleMap
        GMapControl1.Position = New PointLatLng(-12.01921, -77.03346)
        GMapControl1.MinZoom = 0
        GMapControl1.MaxZoom = 24
        GMapControl1.Zoom = 15
        GMapControl1.AutoScroll = True
        'Agregando los poligonos del area de jurisdiccion
        Dim S1 As DataTable
        S1 = New DataTable()
        S1.Columns.Add("")
        S1.Columns.Add("")
        S1.Rows.Add(-12.0299, -77.04297)
        S1.Rows.Add(-12.02264, -77.0504)
        S1.Rows.Add(-12.01092, -77.05281)
        S1.Rows.Add(-12.01051, -77.05035)
        S1.Rows.Add(-12.01091, -77.0496)
        S1.Rows.Add(-12.01183, -77.04925)
        S1.Rows.Add(-12.01238, -77.04929)
        S1.Rows.Add(-12.01273, -77.04879)
        S1.Rows.Add(-12.01382, -77.04905)
        S1.Rows.Add(-12.0145, -77.0488)
        S1.Rows.Add(-12.01471, -77.04902)
        S1.Rows.Add(-12.01546, -77.04847)
        S1.Rows.Add(-12.01548, -77.04803)
        S1.Rows.Add(-12.01573, -77.0475)
        S1.Rows.Add(-12.01621, -77.04715)
        S1.Rows.Add(-12.016, -77.04634)
        S1.Rows.Add(-12.0164, -77.04542)
        S1.Rows.Add(-12.01898, -77.0452)
        S1.Rows.Add(-12.02102, -77.04544)
        S1.Rows.Add(-12.02364, -77.04404)
        S1.Rows.Add(-12.02379, -77.04318)
        S1.Rows.Add(-12.0232, -77.04271)
        S1.Rows.Add(-12.02203, -77.04291)
        S1.Rows.Add(-12.02058, -77.04271)
        S1.Rows.Add(-12.01995, -77.04237)
        S1.Rows.Add(-12.0193, -77.04134)
        S1.Rows.Add(-12.01869, -77.04033)
        S1.Rows.Add(-12.0198, -77.0363)
        S1.Rows.Add(-12.019, -77.03629)
        S1.Rows.Add(-12.01944, -77.03588)
        S1.Rows.Add(-12.01864, -77.03503)
        S1.Rows.Add(-12.02242, -77.03081)
        S1.Rows.Add(-12.02287, -77.03108)
        S1.Rows.Add(-12.02183, -77.03246)
        S1.Rows.Add(-12.02285, -77.03344)
        S1.Rows.Add(-12.02292, -77.03535)
        S1.Rows.Add(-12.02237, -77.03607)
        S1.Rows.Add(-12.02346, -77.03708)
        S1.Rows.Add(-12.02368, -77.03705)
        S1.Rows.Add(-12.0299, -77.04297)
        Dim Poligono_S1 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        Dim lat As Double
        Dim lng As Double
        For i = 0 To S1.Rows.Count - 1
            lat = Convert.ToDouble(S1(i)(0))
            lng = Convert.ToDouble(S1(i)(1))
            puntos.Add(New PointLatLng(lat, lng))
```

```
        Next
        Dim poligonosPunto As GMapPolygon = New GMapPolygon(puntos,
"Poligono")
        Poligono_S1.Polygons.Add(poligonosPunto)
        GMapControl1.Overlays.Add(Poligono_S1)
        Dim S2 As DataTable
        S2 = New DataTable()
        S2.Columns.Add("")
        S2.Columns.Add("")
        S2.Rows.Add(-12.02242, -77.03081)
        S2.Rows.Add(-12.01864, -77.03503)
        S2.Rows.Add(-12.01944, -77.03588)
        S2.Rows.Add(-12.019, -77.03629)
        S2.Rows.Add(-12.01848, -77.03617)
        S2.Rows.Add(-12.01774, -77.03573)
        S2.Rows.Add(-12.01684, -77.0354)
        S2.Rows.Add(-12.01631, -77.03493)
        S2.Rows.Add(-12.01611, -77.03405)
        S2.Rows.Add(-12.01593, -77.03376)
        S2.Rows.Add(-12.01606, -77.03311)
        S2.Rows.Add(-12.01654, -77.03309)
        S2.Rows.Add(-12.01694, -77.03247)
        S2.Rows.Add(-12.01799, -77.03192)
        S2.Rows.Add(-12.0173, -77.03115)
        S2.Rows.Add(-12.02018, -77.02849)
        S2.Rows.Add(-12.02242, -77.03081)
        Dim Poligono_S2 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos2 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S2.Rows.Count - 1
            lat = Convert.ToDouble(S2(i)(0))
            lng = Convert.ToDouble(S2(i)(1))
            puntos2.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto2 As GMapPolygon = New GMapPolygon(puntos2,
"Poligono")
        Poligono_S2.Polygons.Add(poligonosPunto2)
        GMapControl1.Overlays.Add(Poligono_S2)
        Dim S3 As DataTable
        S3 = New DataTable()
        S3.Columns.Add("")
        S3.Columns.Add("")
        S3.Rows.Add(-12.01946, -77.02835)
        S3.Rows.Add(-12.02018, -77.02849)
        S3.Rows.Add(-12.02242, -77.03081)
        S3.Rows.Add(-12.02287, -77.03108)
        S3.Rows.Add(-12.02359, -77.03111)
        S3.Rows.Add(-12.02402, -77.03084)
        S3.Rows.Add(-12.02741, -77.02744)
        S3.Rows.Add(-12.02721, -77.02716)
        S3.Rows.Add(-12.02664, -77.0271)
        S3.Rows.Add(-12.02664, -77.0268)
        S3.Rows.Add(-12.02639, -77.02657)
        S3.Rows.Add(-12.02603, -77.02529)
        S3.Rows.Add(-12.02586, -77.02514)
        S3.Rows.Add(-12.02559, -77.02514)
        S3.Rows.Add(-12.02499, -77.02549)
        S3.Rows.Add(-12.02274, -77.02792)
        S3.Rows.Add(-12.02155, -77.02672)
        S3.Rows.Add(-12.02116, -77.02716)
        S3.Rows.Add(-12.02125, -77.02764)
        S3.Rows.Add(-12.0207, -77.02817)
        S3.Rows.Add(-12.0205, -77.02813)
```

```
        S3.Rows.Add(-12.02008, -77.02792)
        S3.Rows.Add(-12.01963, -77.02808)
        S3.Rows.Add(-12.01946, -77.02835)
        Dim Poligono_S3 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos3 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S3.Rows.Count - 1
            lat = Convert.ToDouble(S3(i)(0))
            lng = Convert.ToDouble(S3(i)(1))
            puntos3.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto3 As GMapPolygon = New GMapPolygon(puntos3,
"Poligono")
        Poligono_S3.Polygons.Add(poligonosPunto3)
        GMapControl1.Overlays.Add(Poligono_S3)
        Dim S4 As DataTable
        S4 = New DataTable()
        S4.Columns.Add("")
        S4.Columns.Add("")
        S4.Rows.Add(-12.01795, -77.02237)
        S4.Rows.Add(-12.01867, -77.0229)
        S4.Rows.Add(-12.01929, -77.02361)
        S4.Rows.Add(-12.01959, -77.02444)
        S4.Rows.Add(-12.01973, -77.02544)
        S4.Rows.Add(-12.01969, -77.02729)
        S4.Rows.Add(-12.01946, -77.02835)
        S4.Rows.Add(-12.01963, -77.02808)
        S4.Rows.Add(-12.02008, -77.02792)
        S4.Rows.Add(-12.0205, -77.02813)
        S4.Rows.Add(-12.0207, -77.02817)
        S4.Rows.Add(-12.02125, -77.02764)
        S4.Rows.Add(-12.02116, -77.02716)
        S4.Rows.Add(-12.02155, -77.02672)
        S4.Rows.Add(-12.02274, -77.02792)
        S4.Rows.Add(-12.02499, -77.02549)
        S4.Rows.Add(-12.02559, -77.02514)
        S4.Rows.Add(-12.02586, -77.02514)
        S4.Rows.Add(-12.02603, -77.02529)
        S4.Rows.Add(-12.02639, -77.02657)
        S4.Rows.Add(-12.02664, -77.0268)
        S4.Rows.Add(-12.02664, -77.0271)
        S4.Rows.Add(-12.02721, -77.02716)
        S4.Rows.Add(-12.02741, -77.02744)
        S4.Rows.Add(-12.02798, -77.02754)
        S4.Rows.Add(-12.02771, -77.02707)
        S4.Rows.Add(-12.02777, -77.0263)
        S4.Rows.Add(-12.02601, -77.02456)
        S4.Rows.Add(-12.02452, -77.02385)
        S4.Rows.Add(-12.02404, -77.02291)
        S4.Rows.Add(-12.02349, -77.0225)
        S4.Rows.Add(-12.02169, -77.02243)
        S4.Rows.Add(-12.02093, -77.02201)
        S4.Rows.Add(-12.02036, -77.0219)
        S4.Rows.Add(-12.01795, -77.02237)
        Dim Poligono_S4 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos4 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S4.Rows.Count - 1
            lat = Convert.ToDouble(S4(i)(0))
            lng = Convert.ToDouble(S4(i)(1))
            puntos4.Add(New PointLatLng(lat, lng))
        Next
```

```
        Dim poligonosPunto4 As GMapPolygon = New GMapPolygon(puntos4,
"Poligono")
        Poligono_S4.Polygons.Add(poligonosPunto4)
        GMapControl1.Overlays.Add(Poligono_S4)
        Dim S5 As DataTable
        S5 = New DataTable()
        S5.Columns.Add("")
        S5.Columns.Add("")
        S5.Rows.Add(-12.01959, -77.02444)
        S5.Rows.Add(-12.01973, -77.02544)
        S5.Rows.Add(-12.01969, -77.02729)
        S5.Rows.Add(-12.01946, -77.02835)
        S5.Rows.Add(-12.02018, -77.02849)
        S5.Rows.Add(-12.0173, -77.03115)
        S5.Rows.Add(-12.01668, -77.03048)
        S5.Rows.Add(-12.0158, -77.02823)
        S5.Rows.Add(-12.01539, -77.02853)
        S5.Rows.Add(-12.0146, -77.02776)
        S5.Rows.Add(-12.01339, -77.0276)
        S5.Rows.Add(-12.01262, -77.02693)
        S5.Rows.Add(-12.01348, -77.02664)
        S5.Rows.Add(-12.01369, -77.02639)
        S5.Rows.Add(-12.01455, -77.02597)
        S5.Rows.Add(-12.01565, -77.02523)
        S5.Rows.Add(-12.01623, -77.02431)
        S5.Rows.Add(-12.01656, -77.02461)
        S5.Rows.Add(-12.01714, -77.02454)
        S5.Rows.Add(-12.01788, -77.02485)
        S5.Rows.Add(-12.01826, -77.02425)
        S5.Rows.Add(-12.01959, -77.02444)
        Dim Poligono_S5 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos5 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S5.Rows.Count - 1
            lat = Convert.ToDouble(S5(i)(0))
            lng = Convert.ToDouble(S5(i)(1))
            puntos5.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto5 As GMapPolygon = New GMapPolygon(puntos5,
"Poligono")
        Poligono_S5.Polygons.Add(poligonosPunto5)
        GMapControl1.Overlays.Add(Poligono_S5)
        Dim S6 As DataTable
        S6 = New DataTable()
        S6.Columns.Add("")
        S6.Columns.Add("")
        S6.Rows.Add(-12.01065, -77.02235)
        S6.Rows.Add(-12.01104, -77.02405)
        S6.Rows.Add(-12.01183, -77.02596)
        S6.Rows.Add(-12.01262, -77.02693)
        S6.Rows.Add(-12.01348, -77.02664)
        S6.Rows.Add(-12.01369, -77.02639)
        S6.Rows.Add(-12.01455, -77.02597)
        S6.Rows.Add(-12.01565, -77.02523)
        S6.Rows.Add(-12.01623, -77.02431)
        S6.Rows.Add(-12.01656, -77.02461)
        S6.Rows.Add(-12.01714, -77.02454)
        S6.Rows.Add(-12.01788, -77.02485)
        S6.Rows.Add(-12.01826, -77.02425)
        S6.Rows.Add(-12.01959, -77.02444)
        S6.Rows.Add(-12.01929, -77.02361)
        S6.Rows.Add(-12.01867, -77.0229)
        S6.Rows.Add(-12.01795, -77.02237)
```

```
        S6.Rows.Add(-12.01585, -77.02235)
        S6.Rows.Add(-12.0153, -77.02253)
        S6.Rows.Add(-12.01457, -77.02242)
        S6.Rows.Add(-12.01392, -77.02246)
        S6.Rows.Add(-12.01294, -77.02209)
        S6.Rows.Add(-12.01065, -77.02235)
        Dim Poligono_S6 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos6 As List(Of PointLatLng) = New List(Of PointLatLng)
        'Variables para almacenar los datos
        For i = 0 To S6.Rows.Count - 1
            lat = Convert.ToDouble(S6(i)(0))
            lng = Convert.ToDouble(S6(i)(1))
            puntos6.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto6 As GMapPolygon = New GMapPolygon(puntos6,
"Poligono")
        Poligono_S6.Polygons.Add(poligonosPunto6)
        GMapControl1.Overlays.Add(Poligono_S6)
        GMapControl1.Zoom = GMapControl1.Zoom + 1
        GMapControl1.Zoom = GMapControl1.Zoom - 1
    End Sub
    Sub Poligono_FA()
        GMapControl1.DragButton = MouseButtons.Left
        GMapControl1.CanDragMap = True
        GMapControl1.MapProvider = GMapProviders.GoogleMap
        GMapControl1.Position = New PointLatLng(-12.01076, -77.03321)
        GMapControl1.MinZoom = 0
        GMapControl1.MaxZoom = 24
        GMapControl1.Zoom = 15
        GMapControl1.AutoScroll = True
        'Agregando los poligonos del area de jurisdiccion
        Dim S1 As DataTable
        S1 = New DataTable()
        S1.Columns.Add("")
        S1.Columns.Add("")
        S1.Rows.Add(-11.99974, -77.03357)
        S1.Rows.Add(-12.00048, -77.03415)
        S1.Rows.Add(-12.00176, -77.03484)
        S1.Rows.Add(-12.00297, -77.03593)
        S1.Rows.Add(-12.00375, -77.03683)
        S1.Rows.Add(-12.00436, -77.0381)
        S1.Rows.Add(-12.00499, -77.03868)
        S1.Rows.Add(-12.00709, -77.03817)
        S1.Rows.Add(-12.00768, -77.03775)
        S1.Rows.Add(-12.0081, -77.03861)
        S1.Rows.Add(-12.00873, -77.03837)
        S1.Rows.Add(-12.00823, -77.03657)
        S1.Rows.Add(-12.00779, -77.03682)
        S1.Rows.Add(-12.00581, -77.03305)
        S1.Rows.Add(-12.00508, -77.03292)
        S1.Rows.Add(-12.00485, -77.03202)
        S1.Rows.Add(-12.00426, -77.03142)
        S1.Rows.Add(-12.0039, -77.03064)
        S1.Rows.Add(-12.00195, -77.03179)
        S1.Rows.Add(-12.00018, -77.03302)
        S1.Rows.Add(-11.99974, -77.03357)
        Dim Poligono_S1 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos As List(Of PointLatLng) = New List(Of PointLatLng)
        'Variables para almacenar los datos
        Dim lat As Double
        Dim lng As Double
        For i = 0 To S1.Rows.Count - 1
            lat = Convert.ToDouble(S1(i)(0))
            lng = Convert.ToDouble(S1(i)(1))
```

```
            puntos.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto As GMapPolygon = New GMapPolygon(puntos,
"Poligono")
        Poligono_S1.Polygons.Add(poligonosPunto)
        GMapControl1.Overlays.Add(Poligono_S1)
        Dim S2 As DataTable
        S2 = New DataTable()
        S2.Columns.Add("")
        S2.Columns.Add("")
        S2.Rows.Add(-12.01319, -77.03716)
        S2.Rows.Add(-12.01209, -77.03725)
        S2.Rows.Add(-12.0118, -77.03678)
        S2.Rows.Add(-12.01003, -77.03803)
        S2.Rows.Add(-12.00873, -77.03837)
        S2.Rows.Add(-12.00823, -77.03657)
        S2.Rows.Add(-12.00779, -77.03682)
        S2.Rows.Add(-12.00581, -77.03305)
        S2.Rows.Add(-12.00508, -77.03292)
        S2.Rows.Add(-12.00485, -77.03202)
        S2.Rows.Add(-12.00426, -77.03142)
        S2.Rows.Add(-12.0039, -77.03064)
        S2.Rows.Add(-12.00546, -77.02991)
        S2.Rows.Add(-12.00747, -77.02734)
        S2.Rows.Add(-12.00898, -77.02807)
        S2.Rows.Add(-12.00995, -77.02811)
        S2.Rows.Add(-12.01201, -77.02871)
        S2.Rows.Add(-12.01226, -77.02859)
        S2.Rows.Add(-12.0123, -77.02932)
        S2.Rows.Add(-12.01264, -77.03031)
        S2.Rows.Add(-12.01231, -77.03088)
        S2.Rows.Add(-12.01228, -77.03155)
        S2.Rows.Add(-12.01263, -77.03188)
        S2.Rows.Add(-12.0131, -77.03275)
        S2.Rows.Add(-12.01291, -77.03672)
        S2.Rows.Add(-12.01319, -77.03716)
        Dim Poligono_S2 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos2 As List(Of PointLatLng) = New List(Of PointLatLng)
        'Variables para almacenar los datos
        For i = 0 To S2.Rows.Count - 1
            lat = Convert.ToDouble(S2(i)(0))
            lng = Convert.ToDouble(S2(i)(1))
            puntos2.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto2 As GMapPolygon = New GMapPolygon(puntos2,
"Poligono")
        Poligono_S2.Polygons.Add(poligonosPunto2)
        GMapControl1.Overlays.Add(Poligono_S2)
        Dim S3 As DataTable
        S3 = New DataTable()
        S3.Columns.Add("")
        S3.Columns.Add("")
        S3.Rows.Add(-12.01814, -77.03565)
        S3.Rows.Add(-12.01709, -77.03589)
        S3.Rows.Add(-12.016, -77.03653)
        S3.Rows.Add(-12.01587, -77.03685)
        S3.Rows.Add(-12.01487, -77.03744)
        S3.Rows.Add(-12.01436, -77.03736)
        S3.Rows.Add(-12.01352, -77.03738)
        S3.Rows.Add(-12.01319, -77.03716)
        S3.Rows.Add(-12.01291, -77.03672)
        S3.Rows.Add(-12.0131, -77.03275)
        S3.Rows.Add(-12.01511, -77.03365)
        S3.Rows.Add(-12.01658, -77.03307)
```

```
        S3.Rows.Add(-12.01702, -77.03237)
        S3.Rows.Add(-12.01795, -77.03187)
        S3.Rows.Add(-12.01827, -77.03234)
        S3.Rows.Add(-12.01778, -77.03256)
        S3.Rows.Add(-12.01734, -77.03303)
        S3.Rows.Add(-12.01661, -77.03357)
        S3.Rows.Add(-12.01652, -77.03443)
        S3.Rows.Add(-12.01684, -77.0349)
        S3.Rows.Add(-12.0173, -77.03524)
        S3.Rows.Add(-12.01814, -77.03565)
        Dim Poligono_S3 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos3 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S3.Rows.Count - 1
            lat = Convert.ToDouble(S3(i)(0))
            lng = Convert.ToDouble(S3(i)(1))
            puntos3.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto3 As GMapPolygon = New GMapPolygon(puntos3,
"Poligono")
        Poligono_S3.Polygons.Add(poligonosPunto3)
        GMapControl1.Overlays.Add(Poligono_S3)
        Dim S4 As DataTable
        S4 = New DataTable()
        S4.Columns.Add("")
        S4.Columns.Add("")
        S4.Rows.Add(-12.01795, -77.03187)
        S4.Rows.Add(-12.01702, -77.03237)
        S4.Rows.Add(-12.01658, -77.03307)
        S4.Rows.Add(-12.01511, -77.03365)
        S4.Rows.Add(-12.0131, -77.03275)
        S4.Rows.Add(-12.01263, -77.03188)
        S4.Rows.Add(-12.01228, -77.03155)
        S4.Rows.Add(-12.01231, -77.03088)
        S4.Rows.Add(-12.01264, -77.03031)
        S4.Rows.Add(-12.0123, -77.02932)
        S4.Rows.Add(-12.01226, -77.02859)
        S4.Rows.Add(-12.01251, -77.02834)
        S4.Rows.Add(-12.01358, -77.02834)
        S4.Rows.Add(-12.01398, -77.02872)
        S4.Rows.Add(-12.01425, -77.02857)
        S4.Rows.Add(-12.01469, -77.02926)
        S4.Rows.Add(-12.01596, -77.02864)
        S4.Rows.Add(-12.01643, -77.02995)
        S4.Rows.Add(-12.01668, -77.03046)
        S4.Rows.Add(-12.01795, -77.03187)
        Dim Poligono_S4 As GMapOverlay = New GMapOverlay("Poligono")
        Dim puntos4 As List(Of PointLatLng) = New List(Of PointLatLng) '--->
se creo una lista
        'Variables para almacenar los datos
        For i = 0 To S4.Rows.Count - 1
            lat = Convert.ToDouble(S4(i)(0))
            lng = Convert.ToDouble(S4(i)(1))
            puntos4.Add(New PointLatLng(lat, lng))
        Next
        Dim poligonosPunto4 As GMapPolygon = New GMapPolygon(puntos4,
"Poligono")
        Poligono_S4.Polygons.Add(poligonosPunto4)
        GMapControl1.Overlays.Add(Poligono_S4)
        GMapControl1.Zoom = GMapControl1.Zoom + 1
        GMapControl1.Zoom = GMapControl1.Zoom - 1
    End Sub
End Class
```

CONSTANCIA DE AUTORIZACION DE USO DE INFORMACION DEL PATRULLAJE VEHICULAR DE LA CPNP PIEDRA LIZA

El Mayor Juan Carlos Paz Oyola, comisario de la CPNP Piedra Liza, hace constar:

Que el Sr. **Alejandro Fabio Salvatore Aylas Barranca,** identificado con DNI N° 70045020, ha recibido información de las actividades del patrullaje vehicular del mes de junio del presente año. Además, cuenta la autorización de usar la información para fines académicos.

Se expide la presente constancia a solicitud del interesado, para fines que estime conveniente.

Lima, 17 de Agosto del 2021

POLICIA NACIONAL DEL PERU
REGION POLICIAL LIMA
DIVTER NORTE 3
COMISARIA PIEDRA LIZA

Mayor Juan Carlos Paz Oyola

CONSTANCIA DE AUTORIZACION DE USO DE INFORMACION DEL PATRULLAJE VEHICULAR DE LA CPNP RIMAC

El Comandante David Edgar Rucana Ccoa, comisario de la CPNP Rimac, hace constar:

Que el Sr. **Alejandro Fabio Salvatore Aylas Barranca,** identificado con DNI N° 70045020, ha recibido información de las actividades del patrullaje vehicular del mes de junio del presente año. Además, cuenta la autorización de usar la información para fines académicos.

Se expide la presente constancia a solicitud del interesado, para fines que estime conveniente.

Lima, 18 de Agosto del 2021

OP. 291943
DAVID E. RUCANA CCOA
COMANDANTE PNP
COMISARIO

Comandante David Edgar Rucana Ccoa

CONSTANCIA DE AUTORIZACION DE USO DE INFORMACION DEL PATRULLAJE VEHICULAR DE LA CPNP MANZANO

El Mayor Jesús Calmell Del Solar Peña, comisario de la CPNP Manzano, hace constar:

Que el Sr. **Alejandro Fabio Salvatore Aylas Barranca,** identificado con DNI N° 70045020, ha recibido información de las actividades del patrullaje vehicular del mes de junio del presente año. Además, cuenta la autorización de usar la información para fines académicos.

Se expide la presente constancia a solicitud del interesado, para fines que estime conveniente.

Lima, 20 de Agosto del 2021

Mayor Jesús Calmell Del Solar Peña

OA-343507
Jesús E. CALMELL DEL SOLAR PEÑA
Mayor PNP
COMISARIO COM PNP EL MANZANO

CONSTANCIA DE AUTORIZACION DE USO DE INFORMACION DEL PATRULLAJE VEHICULAR DE LA CPNP CIUDAD Y CAMPO

El Mayor Antonio Rodriguez Vargas, comisario de la CPNP Ciudad y Campo, hace constar:

Que el Sr. **Alejandro Fabio Salvatore Aylas Barranca,** identificado con DNI N° 70045020, ha recibido información de las actividades del patrullaje vehicular del mes de junio del presente año. Además, cuenta la autorización de usar la información para fines académicos.

Se expide la presente constancia a solicitud del interesado, para fines que estime conveniente.

Lima, 25 de Agosto del 2021

Mayor Antonio Rodriguez Vargas

MARCO A. RODRIGUEZ VARGAS
MAYOR PNP
CIP. 296447

CONSTANCIA DE AUTORIZACION DE USO DE INFORMACION DEL PATRULLAJE VEHICULAR DE LA CPNP FLOR DE AMANCAES

El Capitan Jamil Bristle Lazo Villon, comisario de la CPNP Flor de Amancaes, hace constar:

Que el Sr. **Alejandro Fabio Salvatore Aylas Barranca,** identificado con DNI N° 70045020, ha recibido información de las actividades del patrullaje vehicular del mes de junio del presente año. Además, cuenta la autorización de usar la información para fines académicos.

Se expide la presente constancia a solicitud del interesado, para fines que estime conveniente.

Lima, 25 de Agosto del 2021

POLICIA NACIONAL DEL PERU
COMISARIA FLOR DE AMANCAES
REG. POL. L-DIVTER NORTE 3

Capitán Jamil Bristle Lazo Villon