

UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI



FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES

CARRERA DE COMPUTACIÓN

Tema: "Sistema de gestión de rutas para la recolección de desechos sólidos"

Trabajo de Integración Curricular previo a la obtención del
título de Ingeniero en Ciencias de la Computación

AUTORES: Cadena Castro Estefanía Lizbeth
Lechón Guatemal Jorge Rubén

TUTOR: Ing. Hidalgo Guijarro Jairo Vladimir, MSc.

Tulcán, 2025

CERTIFICADO DEL TUTOR

Certificamos que los estudiantes Cadena Castro Lizbeth Estefanía y Lechón Guatemal Jorge Rubén con el número de cédula 0401992268 y 1728049139 respectivamente han elaborado el trabajo de titulación: "Sistema de gestión de rutas para la recolección de desechos sólidos"

Este trabajo se sujeta a las normas y metodología dispuesta en el Reglamento de la Unidad de Integración Curricular, Titulación e Incorporación de la UPEC, por lo tanto, autorizo la presentación de la sustentación para la calificación respectiva

Ing. Hidalgo Guijarro Jairo Vladimir, MSc.

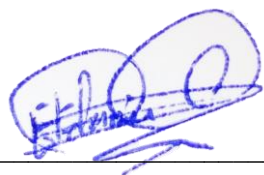
TUTOR

Tulcán, junio de 2025

AUTORÍA DE TRABAJO

El presente Trabajo de Integración Curricular constituye un requisito previo para la obtención del título de Ingeniero en la Carrera de computación de la Facultad de Industrias Agropecuarias y Ciencias Ambientales

Nosotros, Cadena Castro Lizbeth Estefanía y Lechón Guatemal Jorge Rubén con cédula de identidad número 0401992268 y 1728049139 respectivamente declaramos que la investigación es absolutamente original, auténtica, personal y los resultados y conclusiones a los que hemos llegado son de nuestra absoluta responsabilidad.



Cadena Castro Lizbeth Estefanía

AUTORA



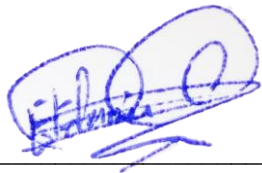
Lechón Guatemal Jorge Rubén

AUTOR

Tulcán, junio de 2025

ACTA DE CESIÓN DE DERECHOS DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Nosotros Cadena Castro Lizbeth Estefanía y Lechón Guatemal Jorge Rubén declaramos ser autores de los criterios emitidos en el Trabajo de Integración Curricular: "Sistema de gestión de rutas para la recolección de desechos sólidos" y se exime expresamente a la Universidad Politécnica Estatal del Carchi y a sus representantes de posibles reclamos o acciones legales.



Cadena Castro Lizbeth Estefanía

AUTORA



Lechón Guatemal Jorge Rubén

AUTOR

Tulcán, junio del 2025

AGRADECIMIENTO

Quiero expresar mi más sincero agradecimiento a la energía superior que ha guiado mi camino con sabiduría, así como a mis ancestros que, desde su ausencia física, han sido luz y fuerza en cada paso de este logro.

A mis padres, mi tía y a mi abuela, pilares fundamentales en mi vida, cuyo amor, entrega y enseñanzas han sido el motor que me impulsó a seguir adelante. Su presencia constante y generosa ha hecho posible cada uno de mis avances, y su apoyo ha sido un refugio en los momentos más difíciles.

Agradezco profundamente al MSc. Jairo Hidalgo, mi tutor, por su guía paciente, su tiempo y su dedicación a lo largo de este proceso. Su conocimiento y compromiso fueron claves para dar forma a esta investigación. Así también, al Municipio de Tulcán y al MSc. Hair Lima Pantoja, jefe de residuos sólidos, por el respaldo institucional y la apertura brindada para el desarrollo de este trabajo. A mi compañero de tesis, Jorge Lechón, gracias por tu esfuerzo, responsabilidad y compañerismo; ha sido valioso compartir esta etapa contigo.

Mi reconocimiento también va a todos los docentes de la carrera, quienes no solo me formaron profesionalmente, sino que también me brindaron herramientas humanas que llevaré por siempre.

Finalmente, a los amigos, amigas, familiares y conocidos, tanto cercanos como distantes, que de manera visible o silenciosa han estado presentes. A ustedes, que saben que estas palabras también son para cada uno, gracias por ser compañía, apoyo y aliento. Cada gesto suyo ha sido un rayo de luz en este camino. Este logro es compartido, y lo guardo con profundo agradecimiento en el corazón.

Estefanía Cadena

Quiero expresar mi más sincero agradecimiento, en primer lugar, a Dios, por darme la fortaleza y la oportunidad de culminar esta importante etapa de mi vida. También agradezco profundamente a mis padres por su constante apoyo en la elección de la carrera de Computación, así como por sus sabios consejos y palabras de aliento, que fueron clave para alcanzar mi meta de titularme. Mi gratitud también se extiende a mi tutor, Msc. Jairo Hidalgo, por su valiosa guía durante el proceso de titulación. Finalmente, agradezco a todos mis amigos, quienes formaron parte de mi formación profesional y contribuyeron a mi crecimiento personal. Gracias a todos

Jorge Lechón

DEDICATORIA

Quiero dedicar este trabajo a quienes han sido pilares esenciales en mi vida y en mi formación académica:

A mi madre, Dolores Castro, por enseñarme el verdadero significado de la resiliencia y a nunca agachar la cabeza ante ningún reto.

A mi padre, Homero Cadena, por compartir conmigo sus valiosos conocimientos y consejos, que han guiado mis pasos y fortalecido mi criterio en cada decisión.

A mi tía, Blanca Castro, por mostrarme el valor de la paciencia y de hacer cada cosa con excelencia.

A mi abuelita, Clemencia Andrade, por enseñarme a afrontar la vida con sabiduría, ya sea desde la paz o con firmeza, según lo requiera la situación.

A mi hermana, Alejandra Villarreal, por estar a mi lado en los momentos más difíciles y ayudarme a sobrellevar el estrés con palabras de aliento y comprensión.

A los miembros de mi familia materna, por su constante preocupación, cariño y respaldo a lo largo de este proceso de titulación.

A mis amigos y conocidos, quienes con su compañía, apoyo y ánimo constante hicieron más llevadero este camino; gracias por compartir risas, consejos y momentos que fortalecieron mi espíritu.

A mi gatita, Cuquito, por acompañarme hasta altas horas de la noche, brindándome su silenciosa compañía y ternura cuando más lo necesitaba.

Cada uno de ustedes, con su amor, enseñanzas y compañía, ha sido parte fundamental de este logro. Esta dedicatoria es para ustedes, con toda mi gratitud y cariño.

Estefanía Cadena

Quiero dedicar este trabajo a mis padres, quienes han sido los pilares fundamentales en mi vida y en mi trayectoria académica. Dedico especialmente esta investigación a mi madre, María Martina Guatemal, y a mi padre, Juan Cecilio Lechón, cuyo incondicional apoyo hizo posible culminar mi etapa de formación profesional. Extiendo también esta dedicatoria a mis hermanos, quienes siempre han estado presentes, guiándome y brindándome su respaldo para alcanzar cada una de mis metas.

Jorge Lechón.

ÍNDICE

RESUMEN	18
ABSTRACT	19
INTRODUCCIÓN	20
I. EL PROBLEMA	21
1.1. PLANTEAMIENTO DEL PROBLEMA	21
1.2. FORMULACIÓN DEL PROBLEMA	22
1.3. JUSTIFICACIÓN	22
1.4. OBJETIVOS Y PREGUNTAS DE INVESTIGACIÓN	23
1.4.1. Objetivo General	23
1.4.2. Objetivos Específicos	23
1.4.3. Preguntas de Investigación	24
II. FUNDAMENTACIÓN TEÓRICA	25
2.1. ANTECEDENTES DE LA INVESTIGACIÓN	25
2.2. MARCO TEÓRICO	28
2.2.1. Sistema de gestión de rutas.....	28
2.2.2. Gestión de Rutas.....	29
2.2.3. Etapas de un sistema de gestión de rutas.....	29
2.2.4. Sistemas de geolocalización	30
2.2.5. Ingeniería de Software	31
2.2.6. Metodologías de desarrollo de software.....	32
2.2.7. Metodología Scrum	36
2.2.8. Normas de calidad del producto de Software.....	38
2.2.9. Lenguajes de Programación	40
2.2.10. Base de datos	44

2.2.11. Base de datos orientada a grafos	46
2.2.12. Teoría de Grafos.....	49
2.2.13. Algoritmos de camino mínimo.....	51
2.2.14. Algoritmo de Dijkstra.....	53
2.2.15. Recolección de desechos Sólidos	58
2.2.16. Proceso de Recolección.....	59
2.2.17. Gestión de Recolección de desechos sólidos	61
2.2.18. Dirección de gestión ambiental y Riesgos	62
2.2.19. COOTAD	62
III. METODOLOGÍA	64
3.1. ENFOQUE METODOLÓGICO	64
3.1.1. Enfoque	64
3.2. IDEA A DEFENDER	65
3.3. DEFINICIÓN Y OPERACIONALIZACIÓN DE LAS VARIABLES	66
3.3.1. Operacionalización variable independiente.....	66
3.3.2. Operacionalización variable dependiente	66
3.4. MÉTODOS UTILIZADOS	66
3.4.1. Métodos.....	66
3.4.2. Técnicas utilizadas	67
3.5. ANÁLISIS ESTADÍSTICO	68
3.5.1. Muestreo probabilístico	68
IV. RESULTADOS Y DISCUSIÓN	69
4.1. RESULTADOS.....	69
4.1.1. Propuesta.....	69
4.1.2. Resultados de la encuesta.....	69
4.1.3. Resultado de la entrevista	77

4.1.4. Fase de Planificación de requerimientos	79
4.1.5. Historias de usuario	83
4.1.6. Fase de diseño de usuario	83
4.1.7. Diagramas de caso de uso.....	85
4.1.8. Diseño de la base de Datos.....	87
4.1.9. Diseño de Interfaces.....	88
4.1.10. Fase de Codificación	93
4.1.11. Presentación del aplicativo	102
4.1.12. Fase de Pruebas.....	107
4.2. DISCUSIÓN	107
V. CONCLUSIONES Y RECOMENDACIONES	109
5.1. CONCLUSIONES.....	109
5.2. RECOMENDACIONES	109
VI. REFERENCIAS BIBLIOGRÁFICAS.....	111
VII. ANEXOS.....	130

ÍNDICE DE TABLAS

Tabla 1. Comparativa de las metodología ágil y tradicional	35
Tabla 2. Comparativas metodologías ágiles	36
Tabla 3. Comparativa lenguajes de programación	41
Tabla 4. Comparativa entre Frameworks.....	43
Tabla 5. Comparativa Base de Datos Relacional	45
Tabla 6. Comparativa de bases de datos orientadas a grafos.....	48
Tabla 7. Tipo de Grafos	51
Tabla 8. Tabla comparativa de algoritmos de camino mínimo	53
Tabla 9. Etapas de la recolección de desechos sólidos.....	60
Tabla 10. Variable independiente "Sistema de rutas"	66
Tabla 11. Variable Dependiente "Recolección desechos sólidos"	66
Tabla 12. Resultado de la Entrevista	77
Tabla 13. Requerimientos funcionales	80
Tabla 14. Requerimientos no funcionales.....	81
Tabla 15. Historia de usuario N°1	83
Tabla 16. Historia de usuario N°2	83
Tabla 17. Historia de usuario N°3	83
Tabla 18. Vehículos utilizados en la recolección de desechos sólidos.....	138
Tabla 19. Tipos de recolección	140
Tabla 20. Actualización de distancias asignadas a cada nodo	152
Tabla 21. Actualización de etiquetas según la nueva iteración.....	153
Tabla 22. Cambio de etiquetas	153
Tabla 23. Etiquetas finales.....	154
Tabla 24. Inicialización del algoritmo de Floyd Warshall.....	155
Tabla 25. Matriz de distancia algoritmo de Floyd Warshall.	156

Tabla 26. Distancia entre i y j de - Floyd Warshall.....	156
Tabla 27. Conexiones existentes - Algoritmo de A*	157
Tabla 28. Nodos y heurística - Algoritmo de A* compilados en una tabla.....	157
Tabla 29. Roles del aplicativo.....	163
Tabla 30. Datos para agregar un nuevo usuario.....	165
Tabla 31. Datos asociados a los vehículos de recolección.....	167
Tabla 32. Opciones desplegadas	179
Tabla 33. Datos al generar una ruta	182
Tabla 34: Términos técnicos.....	189
Tabla 35: Interpretación de código.....	191
Tabla 36: Recepción datos	193
Tabla 37: Respuesta Cliente	197

ÍNDICE DE FIGURAS

Figura 1. Desarrollo de la metodología Scrum.....	37
Figura 2. Matriz de distancia simétrica del grafo G.....	50
Figura 3. Camino óptimo pasando por los puntos 1,3,4,7.	56
Figura 4. Horario de servicio de los camiones recolectores	69
Figura 5. Horario de recolección	70
Figura 6. Días de recolección.....	70
Figura 7. Frecuencia con la que pase el camión recolector.....	71
Figura 8. Tiempo de recolección	71
Figura 9. Conformidad con el horario actual	72
Figura 10. Identificación de problemas la ruta de recolección	72
Figura 11. Puntos de recolección	73
Figura 12. Distancia de puntos de recolección.....	73
Figura 13. Exceso de desechos sólidos	74
Figura 14. Velocidad del servicio de recolección.....	74
Figura 15. Servicio de recolección.....	75
Figura 16. Acumulación de desechos sólidos.....	75
Figura 17. Días de mayor acumulación de desechos sólidos	76
Figura 18. Sugerencia de cambios	76
Figura 19. Gestión de rutas	84
Figura 20. Proceso de recolección de desechos sólidos.....	85
Figura 21. Caso de uso recolección de desechos.....	86
Figura 22. Diseño de la base de datos	87
Figura 23. Prototipo pantalla de inicio.....	88
Figura 24. Prototipo pantalla registro de usuario	88
Figura 25. Prototipo pantalla de recuperación contraseña	89

Figura 26. Prototipo pantalla menú principal	89
Figura 27. Prototipo pantalla agregar información.....	90
Figura 28. Prototipo pantalla agregar nuevo usuario	90
Figura 29. Prototipo pantalla ver información	91
Figura 30. Prototipo pantalla ver información de usuario.....	91
Figura 31. Prototipo pantalla zonas de recolección	92
Figura 32. Prototipo pantalla generar ruta zona 1	92
Figura 33. Código de levantamiento de servidor	93
Figura 34. Código conexión base de datos MySQL.....	94
Figura 35. Código conexión base de datos Neo4j.....	95
Figura 36. Código conexión API Open Street Maps.....	95
Figura 37. Código registro de usuario	96
Figura 38. Código autenticar usuario por contraseña y correo.....	96
Figura 39. Código insertar datos MySQL.....	97
Figura 40. Código consulta visualizar datos MySQL	97
Figura 41. Código editar datos MySQL.....	98
Figura 42. Código eliminar datos MySQL	98
Figura 43. Código autenticación rol administrador	99
Figura 44. Código autenticación rol usuario	100
Figura 45. Código creación relaciones entre puntos de las zonas.....	100
Figura 46. Código creación de rutas.....	101
Figura 47. Código función recursiva	101
Figura 48. Código manejo de prioridades en los puntos.....	102
Figura 49. Interfaz inicial.....	102
Figura 50. Interfaz registro usuario	103
Figura 51. Interfaz recuperación de contraseña.....	103

Figura 52. Interfaz módulos	104
Figura 53. Interfaz módulo Agregar	104
Figura 54. Interfaz módulo información	105
Figura 55. Interfaz registro usuario	105
Figura 56. Interfaz visualizar usuarios	106
Figura 57. Interfaz módulo rutas	106
Figura 58. Interfaz módulo rutas	106
Figura 59. Zona de recolección 1	141
Figura 60. Zona de recolección 2	141
Figura 61. Zona de recolección 3	142
Figura 62. Zona de recolección 4	142
Figura 63. Zona de recolección 5	143
Figura 64. Zona Julio Andrade.....	143
Figura 65. Inicialización de la aplicación del algoritmo de Bellman Ford.....	152
Figura 66. Iteración de A hacia C.....	152
Figura 67. Iteración 2 usando el algoritmo Bellman Ford.....	152
Figura 68. Nueva asignación de distancias	153
Figura 69. Actualización de las etiquetas en la cuarta iteración	153
Figura 70. Grafo final	153
Figura 71. Grafo a aplicar el algoritmo de Floyd Warshall.....	155
Figura 72. Pantalla de inicio sesión	162
Figura 73. Pantalla de registro	163
Figura 74. Pantalla principal menú	163
Figura 75. Menú de navegación.....	164
Figura 76. Pantalla agregar	164
Figura 77. Pantalla Agregar.....	165

Figura 78. Pantalla agregar usuario	166
Figura 79. Pantalla agregar usuario datos necesarios	166
Figura 80. Pantalla agregar vehículo	167
Figura 81. Pantalla agregar vehículo	167
Figura 82. Pantalla agregar horario	168
Figura 83. Pantalla agregar horario opción horario	168
Figura 84. Pantalla agregar ciudad.....	169
Figura 85. Pantalla agregar ciudad.....	169
Figura 86. Pantalla agregar rol	170
Figura 87. Pantalla agregar rol	170
Figura 88. Pantalla modulo eliminar	171
Figura 89. Pantalla eliminar.....	171
Figura 90. Pantalla eliminar usuario.....	172
Figura 91. Pantalla eliminar usuario.....	172
Figura 92. Pantalla eliminar vehículo	173
Figura 93. Pantalla eliminar vehículo con id	173
Figura 94. Pantalla eliminar horario.....	174
Figura 95. Pantalla principal información.....	174
Figura 96. Pantalla información – menú de información.....	175
Figura 97. Pantalla información usuario – selección icono usuario	175
Figura 98. Pantalla información usuario	176
Figura 99. Pantalla información usuario	176
Figura 100. Pantalla información usuario	177
Figura 101. Pantalla información vehículo	177
Figura 102. Pantalla información vehículo	177
Figura 103. Pantalla menú principal	178

Figura 104. Pantalla menú zona de recolección	178
Figura 105. Pantalla zona – visualizar zona 2.....	179
Figura 106. Pantalla zona – Zona 2	180
Figura 107. Pantalla Google maps con coordenadas	180
Figura 108. Excel con puntos de recolección.....	181
Figura 109. Pantalla zona – Botones zona 2	181
Figura 110. Pantalla zona – Limpiar registro de la zona 2	181
Figura 111. Pantalla zona – Registro de ruta vacío	182
Figura 112. Pantalla zona – Generar ruta en la zona 2	183
Figura 113. Pantalla zona 2.....	183
Figura 114. Pantalla zona – Prioridad en el punto 1 de la zona 2	184
Figura 115. Pantalla zona – Eliminar punto 2 de la zona 2.....	184
Figura 116. Pantalla zona – Visualización zona 2.....	185
Figura 117. Zona 2.....	185
Figura 118. Puntos de recolección Zona 2	186
Figura 119. Pantalla del recorrido de la zona 2	186
Figura 120. Visualización de puntos de zona 2.....	187

ÍNDICE DE ANEXOS

Anexo 1. Acta de la sustentación de Predefensa del TIC	130
Anexo 2. Certificado del abstract por parte de idiomas	132
Anexo 3. Entrevista al ingeniero Hair Lima	134
Anexo 4. Distribución de rutas por zona en la ciudad de Tulcán	141
Anexo 5. Cuestionario aplicado en la ciudad de Tulcán.....	144
Anexo 6. Evidencias de la recolección Zona 2.	147
Anexo 7. Aplicación del algoritmo de Bellman Ford	152
Anexo 8. Aplicación del algoritmo Floyd Warshall.....	155
Anexo 9. Aplicación del algoritmo de A*	157
Anexo 10. Certificación de satisfacción	160
Anexo 11. Manual de usuario	161
Anexo 12. Manual Técnico.....	188

RESUMEN

El presente Trabajo de Integración Curricular se centró en la creación de un sistema de gestión de rutas para optimizar la recolección de desechos sólidos. La investigación combinó métodos cuantitativos y cualitativos, incluyendo entrevistas y análisis de procesos, para identificar necesidades clave. Para el diseño y evaluación del sistema, se aplicaron principios de optimización de rutas, buscando una solución adaptable. El sistema se desarrolló con JavaScript y se utilizaron Neo4j y MySQL como bases de datos, permitiendo modelar relaciones geográficas y almacenar datos operativos eficientemente. El análisis detallado de los procesos existentes facilitó la creación de un sistema alineado con las necesidades operativas y geográficas. Se concluyó que la implementación de este sistema tecnológico mejorará significativamente la eficiencia operativa, reducirá costos y optimizará el servicio de recolección.

Palabras Claves: Gestión de Rutas, Optimización, Recolección, Neo4j, JavaScript.

ABSTRACT

This Curricular Integration Work focused on the creation of a route management system to optimize solid waste collection. The research combined quantitative and qualitative methods, including interviews and process analysis, to identify key needs. For the design and evaluation of the system, route optimization principles were applied, seeking an adaptable solution. The system was developed with JavaScript and Neo4j and MySQL were used as databases, allowing to model geographic relationships and store operational data efficiently. The detailed analysis of the existing processes facilitated the creation of a system aligned with the operational and geographic needs. It was concluded that the implementation of this technological system will significantly improve operational efficiency, reduce costs and optimize the collection service.

KEYWORDS: Route Management, Optimization, Collection, Neo4j, JavaScript.

INTRODUCCIÓN

La gestión eficiente de los recursos y la optimización de los procesos son pilares fundamentales en el desarrollo sostenible de las comunidades modernas. En este contexto, la tecnología emerge como una herramienta transversal capaz de transformar diversas áreas, desde la producción industrial hasta la prestación de servicios públicos. Sin embargo, a pesar de los avances tecnológicos, existen problemáticas que aún requieren soluciones innovadoras y adaptadas a las realidades locales. Una de estas problemáticas, de gran impacto ambiental y social, es la gestión de los desechos sólidos urbanos.

La generación de residuos es una constante en las dinámicas sociales y económicas, y su adecuada recolección y disposición representan un desafío significativo para las autoridades locales a nivel global. La eficiencia de los sistemas de recolección influye directamente en la salubridad pública, la calidad del entorno urbano y la sostenibilidad ambiental. En particular, la planificación y la optimización de las rutas de recolección juegan un papel crucial en la eficiencia operativa y la reducción de los impactos negativos asociados a esta actividad esencial.

En el contexto latinoamericano, y específicamente en Ecuador, la gestión de los desechos sólidos presenta particularidades y desafíos que requieren atención. Factores como la infraestructura existente, la planificación urbana y la disponibilidad de recursos influyen en la efectividad de los sistemas de recolección. Por ello, la búsqueda de soluciones tecnológicas que permitan mejorar la eficiencia y la cobertura de estos sistemas se vuelve una necesidad apremiante.

El presente trabajo de investigación se centra en la problemática de la gestión de rutas para la recolección de desechos sólidos, con un enfoque específico en la realidad del Gobierno Autónomo Descentralizado (GAD) Municipal de Tulcán. A través del desarrollo de un sistema de gestión de rutas optimizado, se busca contribuir a una mejora en la eficiencia operativa, la reducción de costos y la optimización del servicio de recolección en esta ciudad.

I. EL PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

En la actualidad, el uso de la tecnología se ha convertido en un factor clave para identificar y optimizar procesos en diversas áreas. No obstante, aún persisten problemáticas sin resolver, como es el caso de la recolección de basura. La generación de desechos sólidos es constante, ocurre en cualquier lugar y como resultado de diversas actividades, lo que provoca una acumulación que representa un problema serio para la comunidad. Frente a esta situación, la responsabilidad de dar solución generalmente recae en los gobiernos autónomos descentralizados, quienes asumen la tarea de gestionar adecuadamente estos residuos. En la mayoría de las regiones, el proceso de recolección sigue un patrón común: los ciudadanos depositan sus desechos en puntos estratégicos dentro de sus barrios, y posteriormente, un camión recolector se encarga de recogerlos para trasladarlos a un centro de tratamiento o a un relleno sanitario.

Según (Sarker y otros, 2021), en países que están en desarrollo, la recolección de basura se ve perjudicada por la carencia de infraestructura y planificación apropiadas. Esto ocasiona una serie de dificultades, como la presencia de rutas inadecuadas, cantidad insuficiente de vehículos y escasez de personal capacitado.

En el ámbito de la planificación, resulta esencial realizar estudios previos que permitan definir rutas de recolección eficientes, establecer horarios adecuados y asegurar una cobertura que responda a las necesidades de la población. Por tanto, se deben promover soluciones orientadas a la reutilización y el reciclaje, con el objetivo de reducir al máximo la cantidad de residuos que llegan a los rellenos sanitarios.

Según (Chinmay, 2024), en naciones en desarrollo, la recolección de desechos sólidos es insuficiente, puesto que enfrentan obstáculos como trayectos inadecuados, falta de transporte y carencia de personal adecuado en la gestión de desechos sólido.

En Latinoamérica, la recolección de desechos sólidos representa un problema común en muchas ciudades de la región. La carencia de rutas óptimas, generan impactos negativos tanto en el entorno urbano como en la calidad de vida de los ciudadanos. Entre las consecuencias más relevantes se destacan la acumulación de basura en espacios públicos, la contaminación ambiental y el incremento de enfermedades asociadas a la insalubridad.

Ecuador, enfrenta desafíos en términos de cobertura y frecuencia, lo que refleja limitaciones en la administración de desechos sólidos. Estas dificultades incluyen los escasos de información, la limitada participación ciudadana y la ausencia de políticas gubernamentales que promuevan una gestión ambiental adecuada. Esta problemática requiere enfoques integrales que permitan implementar soluciones efectivas y sostenibles, con el fin de garantizar un manejo apropiado de los desechos sólidos generados en las ciudades (López y otros, 2023).

En Tulcán, uno de los principales problemas es el uso de un sistema de rutas obsoleto para la recolección de basura. Esta situación genera ineficiencia, ya que los camiones recolectores cubren distancias innecesarias y prolongan su tiempo de operación. Como consecuencia, se presenta insatisfacción ciudadana, cobertura insuficiente y un incremento en las emisiones de gases contaminantes. Además, se ha identificado que los conductores no poseen rutas asignadas, lo que impide cubrir en su totalidad las zonas de la ciudad. Este incumplimiento, también causa embotellamientos, haciendo que el tráfico sea más lento, aumentando los niveles de ruido en la ciudad y provocando la acumulación de basura en diferentes sectores. Como consecuencia, se incrementa el consumo de combustible, debido a recorridos más largos de lo necesario, lo cual repercute negativamente tanto en el tiempo de operación como en el uso de recursos económicos.

1.2. FORMULACIÓN DEL PROBLEMA

¿Cómo un sistema de gestión de rutas contribuirá en recolección de desechos sólidos en la ciudad de Tulcán en el año 2024?

1.3. JUSTIFICACIÓN

La ciudad de Tulcán enfrenta actualmente serias dificultades en la gestión de la recolección de desechos sólidos, producto del uso de un sistema de rutas obsoleto

que genera ineficiencias operativas. Esta problemática se traduce en recorridos innecesariamente largos por parte de los camiones recolectores, lo que provoca un mayor consumo de combustible, incremento en los costos operativos, cobertura deficiente en diferentes zonas, acumulación de basura y, en consecuencia, insatisfacción ciudadana.

Ante esta situación, el desarrollo e implementación de una herramienta tecnológica como "App Rutas" se representa como una solución viable. Este software ha sido diseñado con el objetivo de optimizar las rutas mediante el uso de algoritmos informáticos capaces de calcular trayectorias más eficientes. Su implementación permite reducir el tiempo y la distancia de los recorridos, evitando fallos en la cobertura de las zonas asignadas y facilitando la toma de decisiones basada en datos actualizados.

La propuesta adquiere relevancia dentro del campo de la investigación, ya que se basa en la aplicación de algoritmos de recorrido mínimo adaptados a un contexto particular: el sistema de recolección de desechos sólidos en la ciudad de Tulcán. Esta iniciativa responde a una problemática concreta y actual, respaldada por información real proporcionada por el Departamento de Ambiente del municipio. Además, el trabajo se desarrolla en coordinación con dicha entidad, lo que permite asegurar la pertinencia y factibilidad del proyecto.

En este contexto, el desarrollo de "App Rutas" representa una herramienta estratégica que permite abordar de manera integral los retos de la gestión de residuos sólidos en la ciudad, aprovechando las ventajas de la tecnología.

1.4. OBJETIVOS Y PREGUNTAS DE INVESTIGACIÓN

1.4.1. Objetivo General

- Desarrollar un sistema de rutas óptimas para la recolección de desechos sólidos en la ciudad de Tulcán.

1.4.2. Objetivos Específicos

- Sustentar bibliográficamente las variables de estudio para el desarrollo de un sistema de rutas óptimas enfocado en la recolección basura.

- Investigar sobre el proceso de recolección de desechos sólidos mediante la recopilación cualitativa y cuantitativa de datos en la ciudad de Tulcán.
- Comparar algoritmos informáticos enfocados en la obtención del camino mínimo para el diseño del sistema de rutas óptimas que ayuden en la recolección de basura.
- Diseñar un sistema informático que optimice las rutas del proceso de recolección de desechos sólidos en la ciudad de Tulcán.

1.4.3. Preguntas de Investigación

- ¿De qué manera la fundamentación bibliográfica contribuiría al desarrollo de un sistema eficiente de rutas para la recolección de desechos sólidos en la ciudad de Tulcán?
- ¿Cómo podría un estudio del proceso de recolección de desechos sólidos contribuir a la recopilación de datos cuantitativos y cualitativos en la ciudad de Tulcán?
- ¿Cuál algoritmo informático facilitará el diseño de un sistema de rutas óptimas para la recolección?
- ¿De qué manera un sistema informático puede contribuir a la optimización de las rutas de recolección de desechos sólidos en la ciudad de Tulcán?

II. FUNDAMENTACIÓN TEÓRICA

2.1. ANTECEDENTES DE LA INVESTIGACIÓN

(Castro, 2023) Desarrolló un proyecto denominado "Sistema de ruta óptima y plan de viaje para el sistema de transporte público" el cual está enfocado en el sistema de transporte público para las grandes ciudades. Para la implementación del planificador, se implementó el algoritmo de Dijkstra con el propósito de optimizar las rutas en un sistema informático. La estructura de datos utilizada fue un gráfico dirigido ponderado representado mediante una matriz de adyacencia, ya que el recorrido se realiza en un vehículo terrestre. Dicho grafo modela el mapa del distrito de Chiclayo, incorporando los focos infecciosos y el punto de partida, considerando la orientación de calles y avenidas. Además, para la obtención y visualización de mapas geográficos, se integró el uso de la API V3 de Google Maps.

(Sanchez, 2021) Realizó un artículo científico publicado por la Universidad los Andes, Colombia denominado "Diseño de rutas para la recolección de desechos sólidos implementando herramientas computacionales del sistema de información geográfica (SIG) ArcGIS en un municipio colombiano". La planificación de las rutas de recolección de desechos sólidos, realizada por Juan Esteban Sánchez, constituye un proceso complejo que demanda la consideración de diversas variables, lo que puede complicar la obtención de resultados óptimos. Con el propósito de optimizar los procesos, se han desarrollado herramientas tecnológicas para enfrentar estos retos, destacando especialmente el diseño eficiente de las rutas de recolección. Este proyecto tiene como objetivo principal elaborar un manual metodológico que integra múltiples estrategias para la planificación de recorridos eficientes en la recolección de desechos sólidos, empleando tecnologías geoespaciales como los sistemas de información geográfica (SIG) y herramientas especializadas como ArcGIS, ArcMap y ArcGIS Pro. Como área piloto para aplicar esta metodología, se eligió el municipio de San Juan de Nepomuceno, caracterizado por una población estimada entre 30,000 y 40,000 residentes, una zona urbana delimitada con precisión

y una topografía predominantemente plana, factores que lo convierten en un escenario ideal para el estudio. Se evidenció que la planificación y ejecución de las rutas de recolección de desechos sólidos se optimiza mediante el uso de sistemas de información geográfica, incluso en municipios de tamaño similar o mayor. En conclusión, la planificación de dichas rutas es un proceso complejo que se puede mejorar significativamente con herramientas tecnológicas como los SIG, lo que contribuye a un diseño más eficiente de la recolección selectiva en diferentes tipos de municipios.

(Bravo y otros, 2021) Realizó el proyecto denominado "Planificador de rutas para recojo de desechos sólidos utilizando el algoritmo de Dijkstra" correspondiente a la Revista científica del Señor de Sipán. Con el fin de determinar las rutas más eficientes para la recolección de desechos, se incorporó el algoritmo de Dijkstra, lo que facilitó una segmentación efectiva del área, junto con el uso de los Diagramas de Voronoi. Para modelar el mapa de Chiclayo, se empleó una matriz de adyacencia que representa un grafo dirigido ponderado, considerando las direcciones de las calles y avenidas. Se utilizó la API V3 de Google Maps para que de este modo sea posible la visualización de los mapas geográficos. Se observó una mejora en el tiempo medio de ejecución durante la simulación del planificador que implementó el algoritmo de Dijkstra al trazar rutas con más de 20 puntos de acción. Además, esta investigación se realizó a base de la metodología XP. Los resultados obtenidos en la simulación del planificador de rutas con el algoritmo de Dijkstra mostraron una mejora en el tiempo promedio de ejecución del sistema, especialmente cuando se incrementa el número de focos infecciosos a más de 20.

(López y otros, 2023) llevaron a cabo una investigación denominada "Optimización de rutas en la recolección de residuos tecnológicos". Este análisis se enfocó en la gestión de programación relacionada con el enrutamiento de automóviles, conocido como el problema de enrutamiento de vehículos. En este análisis, se emplearon dos métodos: el algoritmo genético celular y el de optimización inspirado en colonias de hormigas. Los resultados indican que el algoritmo ACO es más eficaz para identificar las rutas más eficientes, logrando así una reducción significativa en la distancia recorrida en comparación con las soluciones generadas por el CGA. Además, se llevaron a cabo análisis estadísticos que respaldan estos hallazgos

(Aragón, 2021) Desarrolló una investigación nombrada como "Optimización de rutas en la recolección de residuos tecnológicos", el cual se centra en un modelo para optimizar rutas accesibles enfocado en personas con discapacidad visual mediante la implementación de un modelo de programación lineal entera y el uso del algoritmo de Dijkstra. Este modelo innovador fue diseñado para minimizar obstáculos, como barreras físicas y esquinas peligrosas, mientras se maximizaban elementos de apoyo esenciales, tales como esquinas seguras, claves ambientales y puntos de ayuda identificados por expertos en movilidad. La investigación incluyó la evaluación de rutas en dos escenarios clave en Lima Metropolitana: el Hospital Arzobispo Loayza y el Hospital María Auxiliadora. Los resultados mostraron que las rutas generadas no solo reducían significativamente las barreras (hasta ocho obstáculos menos) en comparación con las rutas tradicionales, sino que también añadían elementos de apoyo importantes, mejorando así la orientación y seguridad de los usuarios. Aunque las rutas accesibles implicaban un incremento en la distancia (hasta 596 metros adicionales), los beneficios en términos de accesibilidad y seguridad fueron notablemente superiores. Este estudio no solo aporta al desarrollo de rutas seguras para personas con discapacidad visual, sino que también sienta un precedente para implementar soluciones basadas en algoritmos multicriterio en desafíos complejos relacionados con la movilidad urbana. Asimismo, enfatiza la relevancia de tomar en cuenta las necesidades particulares de grupos vulnerables al diseñar infraestructuras urbanas, estableciendo un punto de partida para investigaciones futuras y su aplicación en otros ámbitos, como la planificación y gestión de rutas.

(Talha y otros, 2024) Desarrolló una investigación titulada "Optimal Route Assessment of Oil and Gas Pipelines using Geographic Information System (GIS)". Este estudio fue realizado con el propósito central de identificar e implementar rutas óptimas para la instalación de oleoductos y gasoductos, considerando criterios de eficiencia y seguridad. La investigación se enfocó en la provincia de Khyber Pakhtoon Khwa (KPK), en Pakistán, y buscó reducir los riesgos geológicos que puedan comprometer la integridad de las tuberías a lo largo de su ciclo de vida, desde el diseño hasta la operación. Para alcanzar este objetivo, se integró el uso de Sistemas de Información Geográfica (SIG), herramienta clave para el análisis espacial del terreno y la evaluación de rutas. Esta integración permitió determinar trayectorias que atravesaran terrenos más apropiados, minimizando la exposición a amenazas naturales. La

investigación contrastó las rutas óptimas obtenidas con la red de tuberías existente, priorizando la toma de decisiones fundamentadas en análisis de riesgo y topografía. El estudio reafirma la importancia de una planificación eficiente mediante tecnologías geoespaciales para asegurar la durabilidad y rentabilidad de la infraestructura energética.

(Márquez y otros, 2020) Presentaron un proyecto denominado, "Diseño de rutas de recolección utilizando el algoritmo de optimización por colonia de hormigas". Existe una práctica habitual en la que los vehículos de recolección visitan cada vivienda para retirar los desechos sólidos generados. Sin embargo, las rutas que utilizan estos camiones suelen ser definidas de manera empírica por los conductores, quienes enfrentan desafíos como el tráfico, la escasez de personal y las vías bloqueadas o en reparación, entre otros. Por ello, es fundamental explorar nuevas estrategias que optimicen la recolección de desechos sólidos sin afectar a los ciudadanos, mientras se reducen los tiempos y distancias recorridas en comparación con las rutas tradicionales. Se emplea una metodología basada en técnicas de inteligencia colectiva para diseñar rutas más eficientes, con el propósito de mejorar la gestión de desechos sólidos, reducir los costos operativos y generar trayectos óptimos. En particular, se utiliza la técnica de Optimización por Colonia de Hormigas (ACO, por sus siglas en inglés), la cual se inspira en el comportamiento de las hormigas al buscar alimento. Para ello, se genera un grafo de la zona de interés, que posteriormente se transforma para representar de manera más precisa el problema de ruteo por arcos, considerando las restricciones de la región. Se aplica la técnica ACO sobre este grafo para calcular nuevas rutas. La iniciativa se implementó en el área vial de Villa Milpa Alta, una localidad perteneciente al distrito administrativo de Milpa Alta en la Ciudad de México. Durante el estudio, se contrastaron las rutas propuestas con las implementadas en ese momento, evidenciando la factibilidad de disminuir tanto los kilómetros recorridos como los tiempos de operación de los vehículos recolectores.

2.2. MARCO TEÓRICO

2.2.1. Sistema de gestión de rutas

Un sistema de gestión de rutas es una herramienta tecnológica diseñada para planificar, optimizar y supervisar rutas de transporte, distribución o servicios de manera

eficiente. Este tipo de software analiza múltiples variables como el tráfico, condiciones viales, horarios de entrega, capacidad de los vehículos y prioridades operativas, con el fin de reducir tiempos, costos de combustible y aumentar la productividad. Además de la planificación, estos sistemas suelen incluir funciones complementarias como seguimiento en tiempo real, despacho automatizado y análisis de desempeño, lo que permite a las empresas mejorar continuamente sus operaciones logísticas. En sectores como el comercio electrónico, servicios de campo, salud, distribución de alimentos o gestión de residuos, su implementación resulta clave para garantizar entregas puntuales, reducir impactos ambientales y elevar la satisfacción del cliente (Turton, 2024).

2.2.2. Gestión de Rutas

La gestión de rutas constituye un proceso sistemático orientado a la planificación, optimización y control del movimiento de bienes y mercancías, cuyo objetivo fundamental es maximizar la eficiencia operativa mediante la minimización de costes y tiempos de entrega. Este componente estratégico resulta esencial en el contexto económico contemporáneo, donde la velocidad y precisión representan factores determinantes para la competitividad empresarial en el sector logístico. El proceso comprende diversas etapas interrelacionadas:

- Planificación inicial con recopilación de información pertinente
- Evaluación y selección de rutas considerando múltiples variables
- Optimización mediante herramientas tecnológicas especializadas
- Asignación eficiente a los recursos disponibles
- Monitorización continua durante la ejecución
- Evaluación posterior para identificar oportunidades de mejora.

La incorporación de tecnologías avanzadas como sistemas de geolocalización, software de planificación y análisis de datos en tiempo real ha transformado este campo, permitiendo decisiones más informadas y adaptaciones dinámicas ante circunstancias cambiantes del entorno operativo (Arango, 2024).

2.2.3. Etapas de un sistema de gestión de rutas

Un sistema de gestión de rutas no se limita a trazar recorridos; es una solución integral que cubre todas las fases de la operación logística: planificación, ejecución,

monitoreo y mejora continua. A continuación, se detallan las principales etapas y funciones que caracterizan a un sistema de gestión de rutas moderno y eficiente:

- **Planificación de rutas:** Usa algoritmos predictivos para trazar rutas óptimas considerando datos históricos, condiciones climáticas y demandas futuras. Permite anticiparse a imprevistos y mejorar la eficiencia.
- **Optimización de recursos:** Asigna vehículos y conductores según rutas, prioridad de entrega y consumo de combustible, minimizando costos y maximizando el rendimiento operativo.
- **Adaptación en tiempo real:** Ajusta rutas con base en el tráfico, clima o incidencias. Permite reprogramar entregas, notificar a clientes y optimizar los recorridos al instante.
- **Seguimiento de KPIs:** Controla indicadores como entregas a tiempo, consumo de combustible, paradas por ruta y satisfacción del cliente para evaluar el desempeño.
- **Análisis de datos:** Usa herramientas de visualización y reportes para identificar tendencias, hacer comparaciones entre rutas y realizar análisis de causa raíz.
- **Evaluación de conductores:** Mide desempeño individual con base en puntualidad, precisión en entregas y satisfacción del cliente. Ayuda a detectar fortalezas y necesidades de mejora.
- **Feedback continuo:** Recoge opiniones de clientes y comentarios de los conductores sobre rutas, incidencias o procesos. Fomenta la comunicación y ajustes en las operaciones.
- **Capacitación y desarrollo:** Brinda formación en tecnologías, atención al cliente y conducción eficiente, mejorando la seguridad y la calidad del servicio.
- **Sostenibilidad y nuevas tecnologías:** Integra vehículos eléctricos, rutas verdes y métricas de huella de carbono. Considera también el uso de IA, aprendizaje automático y vehículos autónomos para una gestión más inteligente y ecológica (Woods, 2023).

2.2.4. Sistemas de geolocalización

Son herramientas tecnológicas que permiten conocer la posición geográfica de un dispositivo móvil o cualquier objeto con capacidad de conexión, mediante la

utilización de señales satelitales, redes móviles u otras tecnologías de posicionamiento por lo general se conoce la ubicación de un punto por latitud y longitud.

Las tecnologías de geolocalización son fundamentales en diversos ámbitos, permitiendo determinar la ubicación de dispositivos mediante diferentes sistemas. Estos sistemas varían en su precisión y funcionamiento, adaptándose a distintas necesidades y entornos. A continuación, se resumen los principales tipos de tecnologías de geolocalización:

- **Sistemas de Geolocalización Satelital (GNSS/GPS):** Este método de alta precisión utiliza una red global de satélites para determinar la ubicación exacta de los dispositivos mediante la recepción de sus señales. Es la tecnología principal en navegadores y aplicaciones de localización exterior.
- **Geolocalización por Redes Celulares (GMS):** Esta técnica, menos precisa, se basa en el rastreo de dispositivos a través de las torres de telefonía móvil, identificando la zona general en la que se encuentran. Es esencial para la funcionalidad básica de los teléfonos móviles y puede ofrecer una ubicación aproximada.
- **Geolocalización mediante Redes Inalámbricas (WIFI):** Este sistema aprovecha las señales únicas de las redes WIFI para localizar dispositivos conectados, siendo especialmente útil en interiores donde la señal GPS es limitada. Ofrece una alternativa de localización en entornos cubiertos (Román, 2025).

2.2.5. Ingeniería de Software

Constituye una rama de la ingeniería dedicada a la estructuración y edificación de programas informáticos. Este proceso se fundamenta en la aplicación de metodologías y técnicas rigurosas, análogas a las utilizadas en otras disciplinas de la ingeniería. En su desarrollo, se emplean elementos esenciales como algoritmos, programación, estructuras de datos y documentación técnica, con el fin de crear sistemas informáticos robustos, aplicaciones funcionales y soluciones web eficientes. Su influencia se extiende a diversos ámbitos, desde los complejos sistemas computacionales empresariales hasta las aplicaciones de uso cotidiano y la industria de los videojuegos.

Los objetivos primordiales de la ingeniería de software comprenden:

- Desarrollo y concepción de programas informáticos que se ajusten precisamente a los requerimientos del cliente.
- Gestión y simplificación del desarrollo de programaciones de alta complejidad.
- Creación, diseño, construcción, mantenimiento y administración de sistemas de gestión de datos (bases de datos).
- Supervisión y evaluación del trabajo realizado por equipos técnicos en las áreas de ingeniería de sistemas y redes.
- Resolución de problemas inherentes a plataformas de software ya existentes.
- Optimización de software previamente desarrollado para mejorar su rendimiento y eficiencia (Calvo, 2024).

2.2.6. Metodologías de desarrollo de software

2.2.6.1. Metodologías Tradicionales

Las metodologías tradicionales de gestión de proyectos, también conocidas como metodologías secuenciales, siguen un enfoque lineal que se desarrolla a través de cinco fases principales: iniciación, planificación, ejecución, monitoreo y cierre. Este tipo de gestión se caracteriza por una fuerte planificación inicial, donde se definen el alcance, los requisitos y la documentación desde el principio del proyecto. La responsabilidad central recae en un gerente de proyecto o una Oficina de Gestión de Proyectos (PMO), quienes son los encargados de velar por el cumplimiento de los objetivos. Una vez que el alcance es aprobado, cualquier modificación posterior debe gestionarse a través de un proceso formal de gestión de cambios. Debido a su enfoque estructurado, estas metodologías requieren una carga significativa de trabajo en la fase inicial, permitiendo que las etapas posteriores se desarrollen de manera más fluida (Reaiche, 2022).

Las metodologías tradicionales se caracterizan por su estructura rígida, lineal y poco adaptable a cambios. Son costosas, requieren roles y actividades bien definidos, y demandan documentación extensa durante todo el proyecto. Se centran en seguir un plan acordado inicialmente por todos los involucrados. Entre estas metodologías se encuentran:

- Rational Unified Process: Es una metodología tradicional desarrollada por Rational Software, ahora parte de IBM. Está orientada a proyectos complejos y equipos grandes, promoviendo una gestión disciplinada basada en el lenguaje UML. Aunque tiene estructura rígida, permite cierta adaptabilidad. Integra principios Lean y comparte prácticas con metodologías ágiles como Scrum, permitiendo un desarrollo iterativo e incremental enfocado en la calidad del producto final (Ortega, 2022).
- Microsoft Solutions Framework: Es una metodología ágil desarrollada por Microsoft para gestionar proyectos de software de forma flexible. Se basa en escenarios y contextos reales, promoviendo la adaptación al cambio más que seguir planes estrictos. Incluye flujos de trabajo, roles definidos y actividades clave que guían al equipo a lo largo del ciclo de desarrollo. Su objetivo es facilitar una gestión eficiente y colaborativa, alineada con los principios ágiles (Steinhardt, 2024).
- Modelo en Espiral: Combina elementos de los modelos en cascada, evolutivo y de prototipos. Está diseñado para gestionar proyectos grandes, costosos o complejos, con un enfoque fuerte en la gestión de riesgos. Se estructura en ciclos iterativos que permiten refinar el producto progresivamente. Es ideal cuando se requiere evaluar riesgos continuamente y se trabaja bajo limitaciones de presupuesto o con requisitos complejos y cambiantes (Doshi y otros, 2021).
- ICONIX Process: Es un enfoque de desarrollo de software que combina características del Proceso Unificado Rational (RUP) y de la Programación Extrema (XP). Se basa en el uso de casos de uso y sigue un ciclo de vida iterativo e incremental. Su proceso consta de cuatro fases: análisis de requisitos, análisis y diseño preliminar, diseño detallado e implementación. ICONIX utiliza UML de forma dinámica, enfocándose en diagramas de casos de uso, secuencia y colaboración (Coronado, 2025).
- Método Waterfall o Cascada: Implementa un desarrollo secuencial donde cada fase debe completarse totalmente antes de iniciar la siguiente. Su denominación refleja el flujo descendente del trabajo, similar a una caída de agua, avanzando progresivamente sin la posibilidad de retornar a etapas previas (Laoyan, 2025).

2.2.6.2. Metodologías ágiles

Se basa en el principio de que el cambio es una parte inherente de cualquier proceso. En vez de eludirlo, lo utiliza como una oportunidad estratégica. Este método se compone de un conjunto de prácticas de gestión de proyectos que enfatizan la flexibilidad y la adaptabilidad. El trabajo se distribuye entre equipos multidisciplinarios, lo que permite a las organizaciones actuar con mayor rapidez y responder de manera más eficaz a las demandas de sus clientes. Estrategias enfocadas para la actualidad por su flexibilidad y adaptabilidad de cambios cuando el software está en desarrollo, forma interactiva que garantiza el trabajo en equipo (Laoyan, 2025).

- Scrum: Scrum es un marco de trabajo ágil para gestión de proyectos que permite a los equipos organizar y planificar tareas basándose en valores, principios y prácticas específicas. Inspirado en el rugby, promueve el aprendizaje continuo, la autonomía en la resolución de problemas y la reflexión sobre éxitos y desafíos. Esta metodología está enfocada en maximizar la productividad mediante equipos estructurados que buscan cumplir todos los requerimientos en plazos cortos, destacándose por ser flexible, adaptable y eficiente (Molina, 2024).
- Kanban: Se implementa mediante tableros visuales que simplifican la gestión de proyectos al representar flujos de trabajo y cargas de actividades. En estos tableros, las tareas se organizan en columnas que reflejan las diferentes etapas del proceso, típicamente "Por hacer", "En proceso" y "Terminado". Las tareas, representadas como tarjetas visuales, avanzan progresivamente entre columnas hasta completarse. Este enfoque, ampliamente adoptado por grandes empresas, se enfoca en agilizar los procesos productivos (Martins, 2025).
- Devops: DevOps es una metodología que automatiza tareas que de otro modo serían lentas y repetitivas, liberando a desarrolladores y administradores de sistemas de trabajos tediosos. Su objetivo principal es optimizar sistemas, herramientas y normas para ejecutar acciones automatizadas en respuesta a eventos durante el desarrollo. Integrada por desarrollo de operaciones, seguridad y ciberseguridad, esta metodología rápida busca obtener resultados beneficiosos. Permite la detección automática de problemas en los proyectos, impidiendo su implementación hasta que cumplan con los

estándares requeridos. Además, puede combinarse con otras metodologías ágiles utilizadas por las empresas (García F. , 2023).

A continuación, se muestra una tabla comparativa entre las metodologías ágiles y tradicionales:

Tabla 1. Comparativa de las metodologías ágiles y tradicionales

Características	Metodologías Ágiles	Metodologías Tradicionales
Enfoque	Iterativo e incremental	Secuencial
Equipo	Autónomo y multifuncional	Jerárquico y funcional
Planificación	Basada en la retroalimentación constante	Basada en un plan detallado al comienzo del proyecto
Documentación	Mínima pero suficiente	Completa y detallada
Entrega	Entrega de valor al cliente en ciclos cortos	Entrega de todo el proyecto en una fecha determinada
Adaptabilidad	Puede adaptarse fácilmente a los cambios	Difícil de adaptar a los cambios
Roles	Los roles pueden variar según el proyecto	Roles fijos y definidos
Pruebas	Pruebas continuas y automatizadas	Pruebas finales antes de la entrega
Comunicación	Comunicación continua y efectiva con el cliente	Comunicación limitada con el cliente
Calidad	Se centra en la calidad del producto en todo momento	Se centra en la calidad al final del proyecto

Tras analizar la tabla comparativa anterior entre metodologías ágiles y tradicionales, resulta evidente que las metodologías ágiles ofrecen varias ventajas para nuestro proyecto. Por ello, a continuación, se presenta una tabla comparativa de las principales metodologías ágiles, con el fin de seleccionar la más adecuada.

Tabla 2. Comparativas metodologías ágiles

Parámetros	Devops	Agile	Scrum Lite	Kanban
Roles	Flexibles y colaborativos	Flexibles, pero con definiciones generales	Establece de manera clara los roles: Product Owner, Scrum Master y el Equipo de Desarrollo.	Genéricos y flexibles
Iteraciones	Cortas y sin una duración fija	Cortas, pero sin una duración estrictamente definida	Sprints que duran entre 1 a 2 semanas	No hay iteraciones fijas. El trabajo fluye continuamente
Equipos	Multifuncionales y auto concentrados	Multifuncionales y autoorganizados	multifuncionales y autoorganizados	Flexibles, pero con menor énfasis en la autoorganización
Modificaciones	Se aceptan cambios, pero minimizados.	Se aceptan cambios, pero se busca minimizarlos dentro de las iteraciones	Se permiten cambios durante el sprint, pero se recomienda minimizarlos	Se aceptan cambios en cualquier momento, lo que puede generar interrupciones
Medición	Progreso sin métricas específicas.	Se promueve la medición del progreso, pero sin una métrica específica	Se mide la velocidad del equipo en Puntos de Historia	No se mide directamente la velocidad del equipo
Reuniones y gráficos	Periódicas, comunicación abierta.	Se promueven las reuniones periódicas, pero sin detalles definidos	Planificación, Daily Stand-Up, revisión y retrospectiva	Menos estructuradas y sin detalles definidos

2.2.7. Metodología Scrum

Es un marco de gestión de proyectos que fusiona la filosofía ágil con un enfoque iterativo e incremental. A través de ciclos de trabajo cortos y definidos como 'sprints', equipos multidisciplinares colaboran estrechamente para desarrollar y entregar funcionalidades clave de un producto complejo. Este sistema prioriza la entrega continua a los interesados mediante la adaptación flexible a los cambios y la retroalimentación constante del cliente, diferenciándose de los enfoques tradicionales que buscan una entrega final única y completa (Lomelí, 2023).

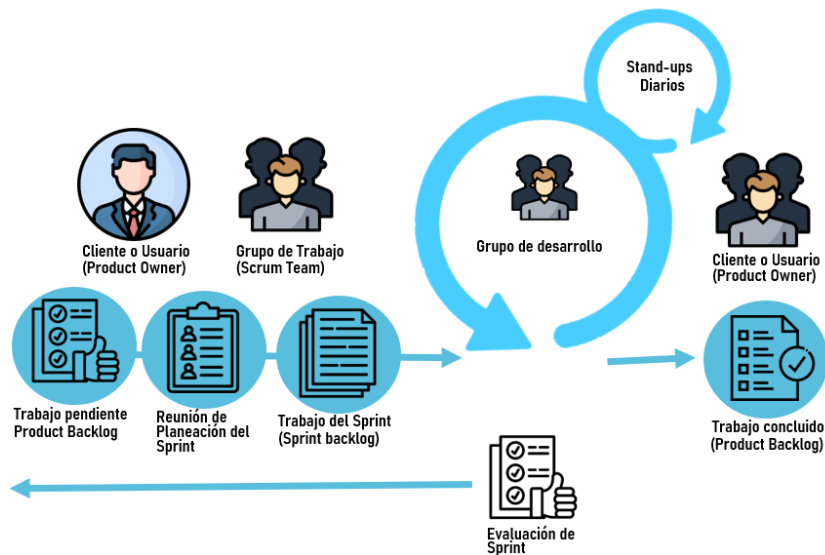


Figura 1. Desarrollo de la metodología Scrum

Fuente: Adaptado de (Lomelí, 2023).

2.2.7.1. Roles de la metodología Scrum

En el marco de la metodología ágil Scrum, se identifican tres roles principales que son fundamentales para la implementación efectiva del proceso. A continuación, se detalla cada uno:

- El Product Owner: Asume la responsabilidad de la gestión y priorización del product backlog, actuando como el nexo entre las necesidades del usuario y el equipo de desarrollo. Su enfoque radica en articular la perspectiva del usuario, clarificando las prioridades de entrega y tomando la decisión final sobre la disposición de los entregables, promoviendo ciclos de entrega frecuentes.
- El Scrum Master: Ejerce como facilitador y gestor del proceso Scrum, liderando los diversos eventos que lo componen. Su rol implica la promoción de las reuniones diarias de actualización, así como la organización y moderación de las sesiones de planificación, revisión y retrospectiva del sprint.
- El Equipo Scrum: Constituye el conjunto de profesionales dedicados a la ejecución del sprint. Se espera que los miembros del equipo operen de manera autoorganizada y colaborativa, con el objetivo común de alcanzar las metas del sprint y fomentar la mejora continua (Lomelí, 2023).

2.2.7.2. Fases de la metodología Scrum

La metodología ágil Scrum se fundamenta en roles y un flujo de trabajo iterativo. Su eficacia radica en la colaboración y la adaptación continua para la entrega de valor. A continuación, se describen los roles principales que estructuran su dinámica operativa.

- Organización del Trabajo Pendiente: Identificación y selección de tareas prioritarias del product backlog por el Scrum Master. Se enfatiza la necesidad de una documentación exhaustiva y centralizada del backlog.
- Planificación del Sprint: Sesión de evaluación y selección de tareas del backlog en las que el equipo se enfocará durante el sprint en curso, definiendo el alcance del trabajo iterativo.
- Inicio del Sprint: Ejecución de las tareas planificadas durante un periodo de tiempo fijo, cuya duración puede ser adaptable a las necesidades del equipo.
- Reuniones de Actualización Diarias (Daily Stand Up): Encuentros breves y cotidianos del equipo de desarrollo para informar sobre el progreso, identificar obstáculos y coordinar las actividades para las siguientes horas.
- Revisión del Sprint: Presentación formal del trabajo completado ("Terminado") a los stakeholders para su aprobación e inspección al finalizar el sprint.
- Retrospectiva del Sprint: Sesión de reflexión y análisis sobre el desarrollo del sprint, donde el equipo identifica aspectos a mejorar para futuros ciclos, promoviendo la mejora continua (Martins, 2025).

2.2.8. Normas de calidad del producto de Software

2.2.8.1. Norma ISO 25000

Surge ante la creciente complejidad y exigencia de calidad en el desarrollo de software impulsado por la rápida evolución de las TIC. A medida que se crean nuevos productos como servicios en la nube o aplicaciones interactivas, se requieren metodologías que garanticen software más robusto, seguro y adaptado a diversos contextos de uso. ISO 25000 propone evaluar aspectos como seguridad, eficiencia y sostenibilidad, promoviendo mejores prácticas para evitar fracasos en los proyectos de TI. Entre sus beneficios se encuentran:

- Establecer y definir requisitos de calidad: Permite identificar tanto necesidades declaradas como implícitas de las partes interesadas. Define la calidad interna, externa y en uso del software, lo que incrementa las probabilidades de éxito del proyecto.
- Medida de calidad y elementos de la medida de calidad: Especifica requisitos de calidad en uso, externos e internos, y permite medir atributos de calidad según las necesidades de todos los involucrados en el proyecto.
- Evaluación de su producto de software: Facilita la evaluación de calidad por parte de desarrolladores, compradores o evaluadores independientes. SQuaRE guía este proceso definiendo entradas, resultados e información clave para asegurar el cumplimiento de los estándares de calidad (Bejtullahu, 2022).

2.2.8.2. Norma ISO 9126

Proporciona un modelo estructurado para evaluar la calidad de productos de software. Este modelo se basa en seis características fundamentales que permiten analizar si un software cumple con los requisitos del usuario, opera eficazmente bajo condiciones específicas y es adaptable a distintos entornos. Las características clave que contempla esta norma son:

- Funcionalidad: Evalúa si el software satisface las funciones requeridas por el usuario y cumple con sus objetivos.
- Confiabilidad: Mide la capacidad del sistema para mantener su rendimiento bajo condiciones específicas por un período determinado.
- Usabilidad: Determina lo fácil que es para los usuarios aprender a usar el sistema y operarlo eficientemente.
- Eficiencia: Analiza el rendimiento del software en relación con los recursos utilizados, como tiempo de respuesta y uso de memoria.
- Mantenibilidad: Considera lo sencillo que resulta identificar, corregir errores o realizar mejoras en el sistema.
- Portabilidad: Evalúa la facilidad con la que el software puede ser transferido o adaptado a diferentes plataformas tecnológicas.

Gracias a esta estructura, la norma ISO 9126 permite establecer criterios claros y objetivos para valorar la calidad del software a lo largo de su ciclo de vida (Tamushi, 2022).

2.2.9. Lenguajes de Programación

Según (Ahamed, 2025) son herramientas fundamentales que permiten a los desarrolladores comunicarse con las computadoras para crear software, resolver problemas complejos y construir soluciones tecnológicas. A medida que surgen nuevas necesidades en sectores como la inteligencia artificial, blockchain, ciencia de datos e Internet de las Cosas (IoT), estos lenguajes evolucionan para adaptarse, volverse más especializados y eficientes. Elegir el lenguaje adecuado puede influir en el rumbo profesional, el tipo de proyectos que se pueden emprender y la capacidad de innovación frente a los desafíos del mundo digital actual.

A continuación, se detallan algunos de los lenguajes de programación más conocidos:

- JavaScript: Es un lenguaje utilizado principalmente para crear páginas web interactivas. Es multiplataforma, lo que permite su ejecución en navegadores y servidores con Node.js. Aunque no es compilado, motores como Chrome V8 optimizan su rendimiento con compilación Just-In-Time (JIT). Es esencial en el desarrollo web, permitiendo funciones interactivas como animaciones y actualizaciones en tiempo real
- Python: Es un lenguaje de programación creado por Guido van Rossum en los años 90. Es utilizado para tareas como administración de sistemas y programación CGI, similar a Perl. Es de código abierto y compatible con la mayoría de las plataformas. Python tiene una sintaxis única, con estructuras de datos como listas, tuplas (listas inmutables) y diccionarios. Además, soporta métodos de listas y diccionarios, así como comprensiones de listas, inspiradas en Haskell.
- PHP: Empezó como una herramienta para rastrear visitantes de su web. En 1995, se lanzó como Personal Home Page Tools. Es un lenguaje de scripting del lado del servidor incrustado en HTML, utilizado para aplicaciones web. PHP genera HTML al ser interpretado en el servidor y es similar a JavaScript en sintaxis

y manejo de arrays. Aunque originalmente no soportaba programación orientada a objetos, esa capacidad se agregó después.

- C#: Se basa en C++, Java, Delphi y Visual BASIC y está diseñado para el desarrollo de software basado en componentes en el .NET Framework, donde se pueden combinar lenguajes como C#, Visual Basic .NET, y F#. Todos usan el Common Type System (CTS) y a su vez se compilan en Intermediate Language (IL), que se convierte a código máquina mediante un compilador Just-in-Time

Tabla 3. Comparativa lenguajes de programación

Características	JavaScript	Python	PHP	C#
Tipo de lenguaje	Interpretado	Interpretado	Interpretado	Compilado (intermedio)
Paradigma	Orientado a objetos, funcional	Multiparadigma	Imperativo, orientado a objetos Similar a C, pero con muchos símbolos	Orientado a objetos
Sintaxis	Similar a C, sencilla	Muy legible y clara		Similar a Java
Rendimiento	Medio	Medio	Medio	Alto (con .NET)
Uso común	Desarrollo web (frontend y backend)	Ciencia de datos, IA, web, automatización	Desarrollo web (backend)	Aplicaciones de escritorio y web
Curva de aprendizaje	Media	Baja (muy amigable para principiantes)	Media	Media
Plataforma	Navegadores, Node.js	Multiplataforma	Servidores web	Windows (principal), multiplataforma con .NET
Soporte	Muy amplia	Muy amplia	Amplia	Muy buena (especialmente en entorno .NET)

2.2.9.1. Frameworks

- Express: Es un framework para aplicaciones web basado en Node.js, creado por TJ Holowaychuk en 2010. Es una herramienta popular, gratuita y ligera que permite desarrollar aplicaciones web usando JavaScript tanto en el cliente como en el servidor, ofreciendo una alternativa a frameworks más pesados como Django o Ruby on Rails. Además, Express.js facilita la creación de rutas y manejo de solicitudes HTTP, siendo ideal para aplicaciones dinámicas. Para lograr aplicaciones escalables y duraderas, es importante diseñar urls limpias y consistentes, evitando detalles técnicos visibles, utilizando una nomenclatura uniforme en la separación de palabras y asegurando que sean fáciles de

comprender, independientemente de las tecnologías subyacentes (Ortuño, 2023).

- **React:** Es un framework basado en JavaScript para desarrollar interfaces de usuario, creado en 2011 por Jordan Walke de Facebook. A pesar de ser un framework, se considera una librería debido a su enfoque en la interfaz de usuario, aunque incluye extensiones como Flux y React Native para soporte completo de aplicaciones. React utiliza Webpack para compilar código y manejar archivos CSS. Además de eso permite crear aplicaciones dinámicas con menos código y mayor funcionalidad que JavaScript tradicional. Usa un Virtual DOM para mejorar el rendimiento, actualizando solo los componentes modificados. Sus componentes reutilizables y flujo de datos unidireccional facilitan el desarrollo y la depuración. La curva de aprendizaje baja y su capacidad para crear aplicaciones web y móviles con React Native explican su popularidad (Deshpande, 2025).
- **Angular:** Es un framework front-end basado en TypeScript, creado y mantenido por Google. Lanzado en 2016 como sucesor de AngularJS, está diseñado para construir aplicaciones web dinámicas de una sola página. Es de código abierto y gratuito, y es utilizado por plataformas como Gmail, Forbes y UpWork. Angular se destaca por su arquitectura basada en componentes, lo que permite crear interfaces de usuario complejas. Aunque enfrenta competencia de nuevas tecnologías, sigue siendo una opción robusta y flexible para el desarrollo de aplicaciones web interactivas (Radkovskyy y Muzyka, 2024).
- **Vue.js:** Es un framework progresivo para construir interfaces de usuario. A diferencia de otros frameworks monolíticos, Vue está diseñado para ser utilizado de manera incremental. Su librería central se enfoca solo en la capa de visualización, lo que facilita su integración con otras librerías o proyectos existentes. Aunque es fácil de empezar, también es capaz de impulsar aplicaciones web sofisticadas cuando se usa con herramientas y librerías adicionales (Vue.js, 2025).
- **Svelte:** Es un framework de interfaz de usuario que utiliza un compilador para permitir a los desarrolladores escribir componentes extremadamente concisos, los cuales realizan una mínima carga de trabajo en el navegador. Este enfoque se basa en el uso de lenguajes bien conocidos como HTML, CSS y JavaScript,

lo que facilita su adopción por parte de los desarrolladores. A diferencia de otros frameworks, Svelte optimiza el rendimiento al trasladar gran parte del trabajo al momento de la compilación, en lugar de hacerlo en tiempo de ejecución, lo que resulta en aplicaciones más rápidas y eficientes. Este framework ha ganado gran popularidad entre los desarrolladores, quienes lo consideran una herramienta innovadora en el desarrollo web. De hecho, múltiples encuestas y estudios destacan a Svelte como uno de los frameworks más emocionantes y prometedores en la actualidad (Svelte.dev, 2025).

A continuación, se muestra un cuadro comparativo entre algunos de los frameworks más usados, juntamente con sus características más destacables

Tabla 4. Comparativa entre Frameworks

Característica	Express	React	Angular	Vue.js	Svelte
Tipo	Framework (Backend)	Librería (Frontend)	Framework (Frontend)	Framework (Frontend)	Framework (Frontend)
Uso Principal	Desarrollo de aplicaciones web del lado del servidor	Desarrollo de interfaces de usuario en aplicaciones web	Desarrollo de aplicaciones web dinámicas de una sola página	Construcción de interfaces de usuario de una sola página de forma progresiva	Creación de interfaces de usuario con un enfoque optimizado de compilación
Lenguaje Principal	JavaScript (Node.js)	JavaScript (JSX)	TypeScript/JavaScript	JavaScript	JavaScript
Facilidad de Aprendizaje	Alta	Moderada	Alta	Alta	Alta
Popularidad	Alta	Muy Alta	Alta	Alta	Alta
Rendimiento	Alto (Especialmente en aplicaciones backend)	Alto (Depende del uso de componentes)	Moderado (Requiere optimización en aplicaciones grandes)	Alto (Escalable y flexible)	Muy Alto (Optimización de trabajo en tiempo de compilación)
Ideal Para	Aplicaciones web completas (Backend + API)	Interfaces interactivas de usuario en aplicaciones web	Aplicaciones web completas y escalables	Aplicaciones web progresivas	Aplicaciones rápidas con mínima carga en el navegador

Express sobresale como la opción más adecuada para el desarrollo de sitios web según el cuadro comparativo. A diferencia de React, Angular, Vue.js y Svelte, que están más orientados a la creación de interfaces de usuario en el frontend, Express se especializa en el desarrollo del backend, proporcionando una estructura ligera y flexible para gestionar solicitudes HTTP y construir APIs eficientes. Mientras que otros frameworks requieren un entorno más complejo y tienen una curva de aprendizaje

más pronunciada, Express permite un desarrollo rápido y escalable, ideal para proyectos que buscan simplicidad y rendimiento en el servidor. Además, su integración con Node.js lo hace altamente compatible con otras herramientas y tecnologías, lo que lo convierte en una opción poderosa y eficiente para aplicaciones web.

2.2.10. Base de datos

Un sistema informático organiza y guarda datos de manera estructurada, gestionados mediante un sistema de gestión de bases de datos (DBMS). Los datos incluyen texto, números y elementos multimedia, y se estructuran en bases de datos. Según (Betania, 2023), los DBMS actúan como intermediarios entre los usuarios y la información almacenada, encargándose de la administración de los datos, motores de datos y esquemas. Una base de datos relacional es una colección de datos organizados en tablas estructuradas, desde las cuales se pueden acceder o reorganizar los datos sin modificar la estructura de las tablas base. A continuación, se detallan algunas bases de datos más usadas:

- MySQL: Según (Gustavo, 2024) es un sistema de gestión de bases de datos relacional de código abierto que sigue el modelo cliente-servidor. Es conocido por su facilidad de uso y flexibilidad, permitiendo modificaciones en su código fuente y una rápida instalación. Su alto rendimiento se logra a través del uso de clústeres de servidores, lo que le permite manejar grandes volúmenes de datos y realizar tareas exigentes. Además, es considerado un estándar en la industria y ofrece una plataforma segura, con funciones de gestión de cuentas, privilegios y cifrado de contraseñas.
- Oracle: Es ampliamente utilizado por organizaciones de todo el mundo, basado en SQL para la manipulación de datos. Es conocido por su alta disponibilidad, escalabilidad y seguridad, soportando big data, análisis predictivos y ofreciendo características como replicación de datos y cifrado (Hayes y Downie, 2024).
- PostgreSQL: PostgreSQL es un sistema de gestión de bases de datos objeto-relacionales de código abierto que organiza datos en tablas, soporta relaciones complejas y permite funciones avanzadas como tipos de datos personalizados, herencia de tablas y compatibilidad con JSON. Es ideal para

manejar grandes volúmenes de datos y realizar consultas y análisis complejos, siendo ampliamente utilizado en proyectos de ciencia de datos e inteligencia artificial (All, 2024).

- **Microsoft SQL:** Microsoft SQL Server es un sistema de gestión de bases de datos relacional ampliamente utilizado en aplicaciones de inteligencia empresarial y análisis en entornos corporativos. Basado en Transact-SQL, ofrece opciones para implementarse tanto en servidores locales como en la nube. Es ideal para almacenar y gestionar grandes volúmenes de datos, como registros de usuarios, catálogos de productos y contenido multimedia, facilitando su administración a través de una interfaz visual. Su motor relacional procesa comandos, consultas y almacena datos, mientras que otros componentes gestionan la memoria y las interacciones con los servidores (Darias, 2021).

Tabla 5. Comparativa Base de Datos Relacional

Características	MySQL	Oracle	PostgreSQL	Microsoft SQL
Tipo	RDBMS de código abierto	RDBMS propietario	RDBMS de código abierto	RDBMS propietario
Licencia	GPL (versión Community), Comercial	Licencia comercial (propietaria)	Licencia PostgreSQL (libre)	Licencia comercial (con ediciones gratuitas limitadas)
Lenguaje consulta	SQL (con algunas extensiones propias)	SQL (con PL/SQL)	SQL estándar + PL/pgSQL	T-SQL (Transact-SQL)
Escalabilidad	Buena	Excelente (muy escalable)	Buena	Muy buena (especialmente con Azure)
Rendimiento	Bueno para cargas medianas	Muy alto para cargas empresariales	Excelente para consultas complejas	Muy alto, especialmente con integración en Windows
Seguridad	Buena (mejora con configuraciones)	Muy robusta (incluye cifrado y auditoría)	Muy buena (seguridad granular)	Excelente (integración con Active Directory)
Uso común	Aplicaciones web, startups	Empresas grandes, banca, ERP	Ciencia de datos, GIS, análisis avanzado	Entornos empresariales .NET, BI
Popularidad	Muy alta	Alta (empresas grandes)	Alta (comunidad académica y técnica)	Muy alta en entornos corporativos

2.2.11. Base de datos orientada a grafos

Una base de datos orientada a grafos es una plataforma especializada para almacenar y manipular datos interconectados, representados en forma de nodos y aristas. Este tipo de base de datos es ideal para gestionar relaciones complejas y asociativas entre los datos. A diferencia de las bases de datos relacionales, que utilizan tablas y claves primarias para vincular información, las bases de datos de grafos utilizan relaciones como elementos fundamentales, permitiendo consultas rápidas y eficientes sobre datos conectados (Pykes, 2024).

Características clave de las bases de datos orientadas a grafos:

- Estructura de grafos
- Consultas rápidas
- Escalabilidad horizontal
- Alto rendimiento
- Facilidad de uso

A continuación, se detallan alguna de las bases de datos orientadas a grafos más reconocidas:

- Neo4j: Es una base de datos NoSQL orientada a grafos, desarrollada en Java. Utiliza nodos para representar objetos y aristas para mostrar las relaciones entre ellos. Su lenguaje de consulta, Cypher, es similar a SQL y permite manipular los datos de forma declarativa. Una de sus principales ventajas es la rapidez en la obtención de datos, incluso con grandes volúmenes de nodos, lo que lo hace ideal para aplicaciones como redes sociales y sistemas de recomendación. Neo4j también ofrece seguridad mediante un sistema de replicación que asegura la persistencia y alta disponibilidad de los datos. Sus arquitecturas, como la High Available Cluster y la Casual Cluster, garantizan la sincronización de los datos y la consistencia mediante el protocolo Raft (Rodón, 2022).
- Amazon Neptune: Es una base de datos de grafos sin servidor, optimizada para alta escalabilidad y disponibilidad, con seguridad integrada y copias de seguridad continuas. Permite replicación de datos entre regiones para baja latencia y recuperación ante desastres. A diferencia de las bases de datos relacionales, Neptune es ideal para manejar datos conectados, siendo perfecta para aplicaciones de publicidad, personalización y análisis en tiempo

real. Su rendimiento superior y flexibilidad de esquemas permiten actualizaciones rápidas y gestión eficiente de perfiles de clientes (Amazon Web Services, 2024).

- ArangoDB: Es una base de datos NoSQL que destaca por su capacidad para manejar estructuras complejas como grafos, documentos y claves/valores dentro de un mismo motor. Este tipo de base de datos fue ideal para un startup que deseaba construir una red social enfocada en conectar inversionistas con startups, analizando visualmente las conexiones de los equipos. Gracias a ArangoDB, se pudieron gestionar eficientemente relaciones dinámicas, búsquedas avanzadas, derechos de acceso personalizados y recomendaciones basadas en fuerza de conexión, todo sin sacrificar el rendimiento (Leremenko, 2020).
- OrientDB: Es una base de datos NoSQL multi-modelo y de código abierto que combina grafos, documentos y clave-valor en una sola plataforma. Permite usar consultas con sintaxis SQL y ofrece flexibilidad con esquemas completos, híbridos o sin esquema. Es ideal para aplicaciones complejas, gracias a sus APIs en Java (Graph, Document y Object) y su capacidad de representar relaciones mediante nodos y conexiones en estructuras tipo grafo (Dassi, 2024).

A continuación, se lleva a cabo una comparativa de las bases de datos orientadas a grafos detalladas anteriormente.

Tabla 6. Comparativa de bases de datos orientadas a grafos

Características	Neo4j	Amazon Neptune	ArangoDB	OrientDB
Tipo de base de datos	Base de datos de grafos	Base de datos de grafos	Base de datos multi-modelo (grafos, documentos, clave-valor)	Base de datos multi-modelo (grafos, documentos, clave-valor, objetos)
Desarrollado en	Java	AWS (Amazon)	C++	Java
Modelo de datos	Grafos (nodos y relaciones)	Grafos (nodos y relaciones)	Grafos, documentos, clave-valor, objetos	Grafos, documentos, clave-valor, objetos
Escalabilidad	Escalabilidad horizontal (cluster)	Escalabilidad automática con replicación en regiones	Escalabilidad horizontal, sharding	Escalabilidad horizontal, replicación multi-maestro
Soporte para ACID	Sí	Sí (con algunas limitaciones)	Sí (con limitaciones en entornos distribuidos)	Sí
Lenguaje de consulta	Cypher (declarativo similar a SQL)	Gremlin y SPARQL	AQL (ArangoDB Query Language)	SQL y Gremlin
Replicación y Clustering	Clustering (solo maestro-esclavo)	Clustering de grafos y replicación global	Replicación y Clustering (CP según el teorema CAP)	Replicación multi-maestro y sharding
Facilidad de uso	Fácil de usar, con mucha documentación y comunidad	Requiere infraestructura AWS y habilidades técnicas	Fácil de usar, especialmente con AQL	Requiere curva de aprendizaje debido a su flexibilidad. Soporta relaciones de grafos, pero también orientado a documentos. Buena integración con otros sistemas de bases de datos.
Manejo de relaciones	Excelente, diseñado específicamente para ello	Bueno, pero más limitado en comparativa	Soporta relaciones de grafos, pero más general	relaciones de grafos, pero también orientado a documentos. Buena integración con otros sistemas de bases de datos.
Integraciones y ecosistema	Integración fuerte con herramientas de IA y análisis	Integración directa con otros servicios AWS	Integración con múltiples modelos de datos y servicios	Integración con otros sistemas de bases de datos.
Mejor opción para grafos puros	Sí (por ser la base de datos de grafos más popular y especializada)	No (aunque adecuado para algunos casos)	No (es más adecuado para datos multi-modelo)	No (menos especializado en grafos puros)
Casos de uso típicos	Redes sociales, recomendaciones, análisis de relaciones	Redes sociales, recomendaciones, IoT, BI	Gestión de datos multidimensionales, análisis de grafos, IoT	Aplicaciones empresariales, análisis de grafos y documentos. Gratuito y de código abierto, con licencias comerciales disponibles.
Costo	Licencia comercial o comunitaria gratuita	Basado en el uso de AWS, puede ser costoso	Gratuito y de código abierto, con licencias comerciales disponibles	Gratuito y de código abierto, con licencias comerciales disponibles

Neo4j se destaca como la mejor opción para bases de datos de grafos puras debido a su enfoque especializado, rendimiento optimizado y una amplia comunidad de soporte. Aunque otras bases como Amazon Neptune, ArangoDB y OrientDB ofrecen características interesantes, como multi-modelo y escalabilidad horizontal, Neo4j es la más adecuada para gestionar relaciones complejas y consultas de grafos con su lenguaje de consulta Cypher. Su facilidad de uso y su ecosistema robusto la posicionan como la opción preferida para aplicaciones centradas en grafos.

2.2.12. Teoría de Grafos

La teoría de grafos es una rama fundamental de las matemáticas que estudia estructuras formadas por nodos (vértices) y las conexiones entre ellos (aristas). Analiza diversos tipos de grafos, como simples, dirigidos, multigrafos, árboles y torneos, así como conceptos clave como recorridos, emparejamiento y flujos en redes. Esta teoría se originó con la resolución del problema de los puentes de Königsberg en el siglo XVIII y ha evolucionado hasta convertirse en una de las áreas más dinámicas de la matemática actual. Abordan temas avanzados como grafos infinitos o aleatorios, mientras que también pueden destacar por su orientación práctica. Incluye aspectos discretos y algebraicos de la teoría, como grafos finitos, resultados de existencia y fórmulas de conteo, y están limitadas tanto por el tiempo disponible en un curso introductorio como por el alcance previsto (Grinberg, 2022).

2.2.12.1. Matrices de Distancia

Una matriz de distancia es una estructura simétrica sin valores negativos, que representa las distancias entre pares de elementos de un conjunto. Se originó en el estudio de grafos conectados, donde cada vértice corresponde a un bucle de comunicación, permitiendo calcular caminos más cortos. Cada elemento de la matriz indica la distancia entre dos vértices, y aunque se reordenen los vértices, su estructura esencial se mantiene. La matriz de distancias es más densa que la de adyacencia, ya que tiene menos ceros fuera de la diagonal, especialmente en grafos densos. A continuación, un ejemplo de la relación existente entre un grafo y una matriz de distancia, en este caso el grafo es G y posee un conjunto de vértices $V(G) = \{V_1, V_2, \dots, V_n\}$ y d_{ij} representa la longitud del camino más corto entre los vértices V_i y V_j . Entonces, definimos la matriz de distancias de G con el conjunto de vértices

denotada por $D(G)$ como una matriz de $n \times n$ cuya entrada es d_{ij} . Esta matriz es real, simétrica y también tiene una traza igual a cero (Bock, 2025).

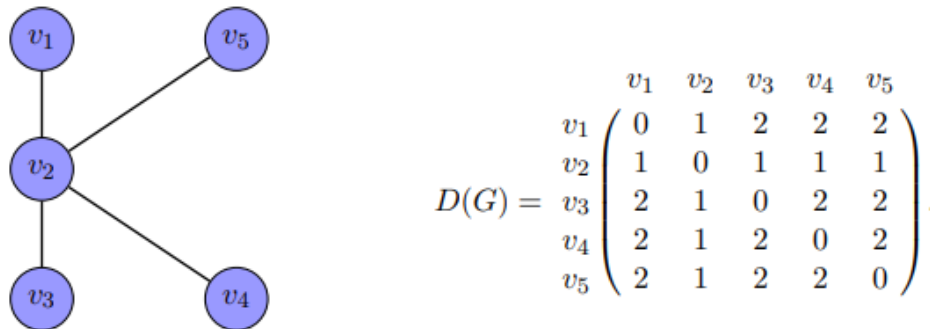


Figura 2. Matriz de distancia simétrica del grafo G
Fuente: Extraído de (Dadedzi y Wagner, 2014).

2.2.12.2. Conceptualización y Tipos de Grafos

Según (Ravikiran, 2025) son estructuras de datos no lineales formadas por nodos (vértices) y conexiones (aristas). Representan relaciones entre elementos, siendo ideales para modelar redes del mundo real como redes telefónicas, circuitos o redes sociales. Cada nodo puede simbolizar una entidad, como un usuario, y las conexiones representan relaciones entre ellos, como llamadas telefónicas o amistades

Grafo dirigido: También conocidos como dígrafos, surgen al convertir las aristas en aristas de sentido único. Siendo así que los dígrafos se identifican por poseer flechas que indican un único sentido. En estos grafos es posible el trazo de caminos, donde un camino se define como una manera de ir de un vértice a otro y consiste en una secuencia de aristas, una luego de otra. De manera formalizada como un par ordenado $G = (V, E)$. V es un grupo finito de vértices y E es un grupo de aristas (o bordes), donde cada arista es un par ordenado de vértices, es decir, $E \subseteq V \times V$ (Fuchs y Ntokos, 2023).

- Grafo no dirigido: Se define como una estructura matemática formada por un conjunto de vértices (o nodos) y un conjunto de aristas (o bordes) que enlazan pares de vértices. En este tipo de grafo, las aristas son no dirigidas, esto significa que, si hay una arista que une el nodo A con el nodo B, se asume igualmente que B está conectado con el nodo A. Esto implica que las relaciones representadas son simétricas. De manera formalizada como un par ordenado $G = (V, E)$. Donde V es un conjunto finito de vértices o nodos. Así E es un

conjunto de aristas, donde sus subconjuntos de V de cardinalidad dos. Siendo representado por $E \subseteq \{\{u, v\} | u, v \in V\}$ (Fuchs y Ntokos, 2023).

- Grafos ponderados: Un grafo ponderado es aquel en el que las aristas tienen valores asignados, llamados pesos. El gráfico ponderado tiene relevancia en el problema de la ruta más corta, que busca encontrar el camino con el menor número de enlaces y el menor peso total. En un grafo ponderado, la ruta más corta podría tener más enlaces que en un grafo no ponderado, ya que el peso de las aristas influye en la selección de la mejor ruta (Ghosh y otros, 2024).
- Grafos no ponderados: Un grafo no ponderado es una representación matemática en la que las aristas solo indican la existencia de una conexión entre nodos, sin asociarles ningún valor numérico o peso. En este tipo de grafos, lo importante es saber si dos vértices están conectados o no, sin considerar factores como distancia, costo o capacidad. Cuando existe una arista entre dos nodos, se dice que son adyacentes o vecinos. Este modelo es útil en situaciones donde la relación entre elementos es meramente binaria, como en redes sociales básicas, conexiones de red o estructuras organizativas (Simic, 2024).

Tabla 7. Tipo de Grafos

Características	Grafo dirigido	Grafo no dirigido	Grafo ponderado	Grafo no ponderado
Dirección	Tiene dirección (→)	Sin dirección (—)	Dirigido o no dirigido	Dirigido o no dirigido
Peso	Puede tener peso	Puede tener peso	Sí	No
Ciclos	Puede ser cíclico o no	Puede ser cíclico o no	Puede ser cíclico o no	Puede ser cíclico o no
Conectividad	Puede ser conexo o no	Puede ser conexo o no	Puede ser conexo o no	Puede ser conexo o no
Ejemplo común	Redes sociales (seguidores), flujos de datos	Redes de amistad, redes de transporte	Mapas de rutas con distancias	Relaciones binarias simples

2.2.13. Algoritmos de camino mínimo

- Dijkstra: Este enfoque forma parte de los algoritmos de búsqueda y se especializa en identificar el camino más corto desde un nodo inicial hacia otros dentro de una red. Su funcionamiento implica realizar iteraciones repetidas, lo que aumenta la complejidad de su ejecución conforme se expande la red,

haciendo que sea menos eficiente en comparación con otros métodos de optimización basados en programación matemática (Bagus, 2020).

- Bellman – Ford: El algoritmo, cuyos desarrolladores corresponden a Richard Bellman, Samuel End y Lester Ford, determina la ruta óptima en un grafo dirigido con pesos, incluso cuando algunas aristas tienen valores negativos. Aunque el algoritmo de Dijkstra aborda el mismo problema con mayor rapidez, exige pesos positivos en las aristas, lo que hace que el algoritmo mencionado sea preferido cuando estas incluyen valores negativos (Mukhlif y Saif, 2020).
- Algoritmo de Floyd – Warshall: Se trata de un procedimiento utilizado para calcular las distancias mínimas entre todos los pares de nodos en un grafo ponderado (no dirigido). Este algoritmo es capaz de manejar pesos negativos, siempre y cuando no existan ciclos con peso negativo. El grafo se representa como una matriz de adyacencia $dist$, donde $dist[i][j]$ representa el valor asociado a la arista que une los nodos i y j . Si no hay conexión directa entre los nodos i y j , $dist[i][j]$ se inicializa con un valor muy grande (representando infinito) Y $dist[i][i]$ mantuvo como valor inicial 0 porque la distancia desde un nodo a sí mismo es siempre 0 (Anil, 2024).
- Algoritmo A*: Búsqueda informada o inteligente que encuentra determina la ruta más eficiente desde un punto de partida hasta el objetivo dentro de un espacio de problemas empleando una heurística adecuada. No obstante, en algunos casos, podría dejar de lado pasos más directos, generando una solución óptima. Un algoritmo de búsqueda admisible es aquel que garantiza encontrar una ruta óptima entre el nodo inicial y el nodo objetivo, si es que existe. En la búsqueda A*, se considera que una heurística es válida cuando no supera la estimación de la distancia restante entre el nodo actual y el objetivo. Por ejemplo, la distancia real entre dos ciudades siempre será igual o mayor equivalente a la distancia en línea recta que se observa en un mapa, esta medida es considerada una heurística válida, ya que es "optimista", lo que cumple con la definición (Putri, 2024).

Tabla 8. Tabla comparativa de algoritmos de camino mínimo

Algoritmo	Complejidad temporal	Complejidad espacial	Escalabilidad	Tipo de grafo soportado	Uso típico	Heurística requerida
Dijkstra	con cola de prioridad $O((V + E)\log V)$	Alta (cola de prioridad)	Ideal para grafos pequeños/medianos	Sin pesos negativos	Rutas más cortas desde un solo origen	No
Floyd-Warshall	$O(V^3)$	Muy alta (matriz)	Escalabilidad limitada a grafos pequeños	Pesos positivos o negativos	Comparación entre todas las parejas de nodos	No
Bellman-Ford	$O(V)$	Media	Bueno para grafos medianos	Sin ciclos negativos	Rutas más cortas desde un nodo, detecta ciclos negativos	No
A*	Depende del heurístico, mejor caso	Variable (según implementación)	Muy eficiente con buenos heurísticos	Sin restricciones (costo depende del diseño)	Rutas más cortas optimizadas con heurísticas	Sí

Gracias a su eficiencia y adaptabilidad en grafos con pesos positivos, el algoritmo de Dijkstra es perfecto para diseñar sistemas de rutas óptimas, siendo una herramienta sólida para resolver problemas de logística y navegación. Su complejidad temporal, especialmente cuando se utiliza una cola de prioridad, permite procesar grafos de tamaño pequeño a mediano con rapidez, asegurando rutas más cortas desde un solo origen hacia todos los demás nodos. Además, no requiere una heurística, lo que simplifica su implementación y lo hace altamente confiable en escenarios donde la precisión y la consistencia son prioritarias. Por estas características, Dijkstra es una opción ampliamente adoptada en aplicaciones como GPS, planificación de redes y gestión de tráfico, garantizando rutas óptimas de manera eficiente y escalable.

2.2.14. Algoritmo de Dijkstra

Según (Putri, ResearchGate, 2024) es una herramienta fundamental y eficiente para encontrar la ruta más corta entre dos vértices en un grafo con pesos positivos. Su eficiencia algorítmica lo ha hecho ampliamente aplicable en áreas como redes informáticas, planificación de transporte, sistemas de navegación y logística.

Además, su adaptación a entornos dinámicos y su combinación con tecnologías modernas como la inteligencia artificial y la computación en la nube permiten enfrentar desafíos actuales y futuros en la optimización inteligente de rutas y caminos complejos.

- **Inicialización:** Define un valor inicial asignando una distancia infinita a todos los nodos, excepto al nodo inicial, cuya distancia se establece en 0. Utiliza una estructura de datos, como un montículo o una cola de prioridad, para organizar los nodos que aún están pendientes de visitar.
- **Selección del nodo:** Selecciona el nodo no visitado que tiene la distancia más baja registrada. Este nodo se considera como "visitado" y queda excluido de futuros procesamientos.
- **Actualización de distancias:** Examina los vecinos no visitados a partir del nodo escogido, determina la distancia acumulada desde el nodo de origen incluyendo este nodo. Si el valor calculado es inferior a la distancia previamente registrada para ese vecino, se actualiza con la nueva distancia.
- **Repetición:** Continúa ejecutando los pasos 2 y 3 hasta que no queden nodos pendientes por visitar.
- **Finalización:** El proceso concluye cuando se han explorado todos los puntos y se han calculado las distancias más cortas desde el punto inicial hasta todos los demás.

Este algoritmo recibe su nombre en honor al doctor Edsger W. Dijkstra, quien lo descubrió. Está diseñado para resolver el desafío de determinar la ruta más eficiente, desde un punto en un grafo (la fuente) hasta un destino. Es capaz de identificar las rutas más cortas desde un punto de origen a todos los nodos en un grafo en el mismo tiempo, por lo que a menudo se le conoce como la cuestión de encontrar las rutas más cortas desde un único origen

Aplicado a la vida real, este algoritmo puede ser de gran utilidad, dado que existen muchas estructuras que pueden representarse mediante grafos, y muchos problemas pueden resolverse con la ayuda de ellos. La red de carreteras en un área puede representarse mediante un grafo. Aunque su comprensión resulta mucho más sencilla a través de un ejemplo. Es así como tomando un grafo G el

cual es definido como un par ordenado de (V, E) y es denotado por $G = (V, E)$, donde V y E son definidos como $V = \{V_1, V_2, \dots, V_n\}$ un grupo finito no vacío cuyos miembros se llaman vértices. Mientras que $E = \{e_1, e_2, \dots, e_n\}$ es el conjunto de aristas que conecta un par de vértices.

Los dos vértices (u, v) se consideran adyacentes, siempre que haya una arista que enlace el vértice u con el vértice v . Pero si e es la arista que conecta el vértice u con el vértice v , entonces e se puede escribir como $e = (u, v)$, entonces u y v se consideran localizados en e y se dice que e es incidente con los vértices u y v

$$V(G): \{V_1, V_2, V_3, \dots, V_n\}$$

L es grupo de vértices permanentes (el grupo de vértices $V(G)$) que han sido seleccionados en el camino óptimo.

$D(V_j)$ es la suma de los pesos del camino óptimo de V_i a V_j

$W(V_i, V_j)$ corresponde a los pesos laterales de los vértices V_i a V_j

$W * (1, V_j)$ es la suma de los pesos del camino óptimo de V_i a V_j

Resolver un grafo por el algoritmo Dijkstra.

- Inicialización $L = \{1\}$; $V = \{V_1, V_2, V_3, \dots, V_n\}$
- Para $i = 2, \dots, n$ se aplica $D(V_i) = W(V_i, V_1)$
- Durante $V_n \notin L$ (V_n no es un punto permanente), o: se selecciona un vértice $V_k \in V - L$ con el valor más bajo $D(V_k)$, entonces $L = L \cup \{V_k\}$, o existe la opción de que por cada $V_j \in V - L$ se aplica: si $D(j) > D(k) + W(k, j)$ entonces se reemplaza $D(j)$ con $D(k) + W(k, j)$
- Para cada $V_j \in V$, $W * (1, j) = D(j)$

Según el algoritmo anterior, el camino más corto desde el vértice V_1 hasta V_n es a través de los vértices en L secuencialmente y la suma ponderada de los más cortos es (V_n)

Por tanto la distancia total de una de las rutas es la suma de los pesos que se encuentran continuas $Largo(p) = \sum_{i=1}^k w(V_{i-1}, V_i)$

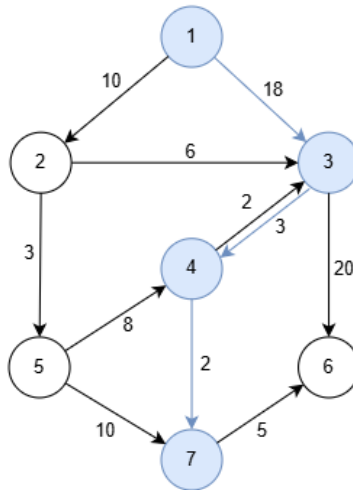


Figura 3. Camino óptimo pasando por los puntos 1,3,4,7.

- Inicialización del grafo propuesto, figura 4.

Conjunto de vértices $V(G) = \{1,2,3,4,5,6,7\}$

Conjunto de vértices permanentes $L = \{\}$

Distancias iniciales desde V_1

$D(1) = 0$ distancia desde el nodo fuente,

$$D(2) = \infty, \quad D(3) = \infty, \quad D(4) = \infty, \quad D(5) = \infty, \quad D(6) = \infty, \quad D(7) = \infty$$

- Iteración 1 Seleccionar V_1 a L

$$L = \{1\}$$

Se actualizan las distancias para los vecinos de V_1

$$D(2) = \min(\infty, D(1) + W(1,2)) = \min(\infty, 0 + 10) = 10$$

$$D(3) = \min(\infty, D(1) + W(1,3)) = \min(\infty, 0 + 3) = 3$$

Distancias actuales

$$D(1) = 0, D(2) = 10, D(3) = 3,$$

$$D(4) = \infty, \quad D(5) = \infty, \quad D(6) = \infty, \quad D(7) = \infty$$

- Iteración 2 Seleccionar $V_3(D(3) = 3)$

Se añade V_3 a L $L = \{1,3\}$

Actualizamos las distancias para los vecinos de V_3

$$D(4) = \min(\infty, D(3) + W(3,4)) = \min(\infty, 3 + 3) = 6$$

$$D(6) = \min(\infty, D(3) + W(3,6)) = \min(\infty, 3 + 20) = 23$$

Distancias actuales

$$D(1) = 0, D(2) = 10, D(3) = 3, D(4) = 6, D(5) = \infty$$

$$D(6) = 23, D(7) = \infty$$

- Iteración 3 Seleccionar $V_4(D(4) = 6)$

Añadimos V_4 a L $L = \{1,3,4\}$

Actualizamos las distancias para los vecinos de V_4

$$D(3) = \min(3, D(4) + W(4,3)) = \min(3, 6 + 2) = 3(\text{nocambia})$$

$$D(7) = \min(\infty, D(4) + W(4,7)) = \min(\infty, 6 + 2) = 8$$

Distancias actuales

$$D(1) = 0, D(2) = 10, D(3) = 3, D(4) = 6,$$

$$D(5) = \infty, D(6) = 23, D(7) = 8$$

- Iteración 4 Seleccionar $V_7(D(7) = 8)$

Añadimos V_7 a L $L = \{1,3,4,7\}$

Actualizamos las distancias para los vecinos de V_7

$$D(6) = \min(23, D(7) + W(7,6)) = \min(23, 8 + 5) = 13$$

Distancias actuales

$$D(1) = 0, D(2) = 10, D(3) = 3, D(4) = 6,$$

$$D(5) = \infty, D(6) = 13, D(7) = 8$$

- Iteración 5 Seleccionar $V_2(D(2) = 10)$

Añadimos V_2 a L $L = \{1,3,4,7,2\}$

Actualizamos las distancias para los vecinos de V_2

$$D(5) = \min(5, D(2) + W(2,5)) = \min(5, 10 + 3) = 5$$

$$D(3) = \min(3, D(2) + W(2,3)) = \min(3, 10 + 6) = 3(\text{nocambia})$$

Diferencias actuales

$$D(1) = 0, D(2) = 10, D(3) = 3, D(4) = 6$$

$$D(5) = 13, \quad D(6) = 13, \quad D(7) = 8$$

- Iteración 6 Seleccionar V_5 ($D(5) = 13$)

Añadimos $V_5 = L$ $L = \{1,3,4,7,2,5\}$

Actualizamos las distancias para los vecinos de V_5

$$D(4) = \min(6, D(5) + W(5,4)) = \min(6, 13 + 8) = 6(\text{no cambia})$$

$$D(7) = \min(8, D(5) + W(5,7)) = \min(8, 13 + 10) = 8(\text{no cambia})$$

Distancias actuales

$$D(1) = 0, \quad D(2) = 10, \quad D(3) = 3, \quad D(4) = 6,$$

$$D(5) = 13, \quad D(6) = 13, \quad D(7) = 8$$

- Iteración 7: Seleccionar V_6 ($D(6) = 13$)

Añadimos V_6 a L $L = \{1,3,4,7,2,5,6\}$

No hay mas vecinos que necesiten actualización

- Resultado final

El camino más corto desde V_1 hasta V_7 es $V_1 \rightarrow V_3 \rightarrow V_4 \rightarrow V_7$

Distancia total $D(1,7) = D(1,3) + D(3,4) + D(4,7) = 18 + 3 + 2$

$$D(1,7) = 23$$

2.2.15. Recolección de desechos Sólidos

Los desechos sólidos son desechos sólidos producidos diariamente por las personas, y se distinguen por su estado sólido y su naturaleza no biodegradable, lo que la diferencia de los desechos sólidos líquidos o gaseosos. Estos materiales son considerados basura, ya que generalmente se descartan en vertederos una vez que se presume que han completado su vida útil. La recolección de desechos sólidos constituye una etapa esencial en la gestión integral de residuos, impactando directamente la salubridad urbana, la prevención de enfermedades y la preservación ambiental. Es así como este proceso abarca la captación, el almacenamiento temporal, el transporte y la disposición final de los residuos generados en diversos sectores. La planificación del servicio de recolección debe

priorizar la periodicidad, la tipología de desecho, el volumen generado y las áreas de cobertura, elementos cruciales para asegurar una operatividad eficiente y ambientalmente responsable (Reyna, 2021).

2.2.16. Proceso de Recolección

El manejo adecuado de los desechos sólidos implica diversas actividades, desde su generación hasta su disposición final. Cada etapa requiere la participación de ciudadanos y trabajadores municipales, con controles y verificaciones para asegurar su correcta gestión. A continuación, se presenta un cuadro comparativo con las actividades clave en este proceso, destacando los responsables, puntos de control y medios de verificación.

Tabla 9. Etapas de la recolección de desechos sólidos

Actividad	Descripción de la actividad	Proceso	Responsables	Punto de control	Medio de verificación
Generación de desechos sólidos	Se generan desechos sólidos como: tarrinas de comida, botellas plásticas, bolsas plásticas	Toda actividad genera desechos sólidos lo cual se acumula y resulta un porcentaje significativo. Clasificación o separación de la basura	Ciudadanos	Hogares, comercios e Industrias	Fotografías
Almacenar temporalmente los desechos sólidos generados.	Control de los desechos sólidos	por tipo; orgánico e inorgánico, colocación en contenedores diferentes.	Ciudadanos e Trabajadores	Hogares, comercios e Industrias	Fotografías Registro de cantidades generadas.
Transportar los desechos comunes acopiados hasta la ciudad para que el recolector los recoja	Los desechos sólidos que han sido depositados temporalmente en contenedores o puntos estratégicos de la localidad	La entidad encargada en la recolección gestionara el recojo de los desechos sólidos para transportarlo Los carros recolectores una vez de obtener el total de una zona	Trabajadores del Municipio de la ciudad del departamento ambiente.	Ubicación de contenedores	Fotografías
Disposición final de los desechos sólidos	Traslado de los desechos hasta el relleno sanitario	específica termina la operación en un punto específico denominado relleno sanitario.	Trabajadores del Municipio de la ciudad del departamento ambiente.	Relleno sanitario de la ciudad	Fotografías Informes

2.2.17. Gestión de Recolección de desechos sólidos

La participación de los ciudadanos y la obligación de proporcionar servicios públicos, incluyendo educación, salud y gestión de desechos sólidos. (Ambientum, 2022) describe este proceso en varios pasos: la prerrecogida en los hogares, la recogida que implica el transporte de los desechos sólidos desde la ciudad a un relleno sanitario, y el tratamiento final, Garantiza una administración eficaz y que respeta el entorno natural.

- Prerrecogida

En esta etapa se llevan a cabo las acciones posteriores a la generación de un residuo, como su separación, manejo y almacenamiento, generalmente realizadas en los hogares (Grupo Sulo, 2021).

- Recogida

Esta etapa consiste en trasladar los desechos sólidos desde los hogares hasta un sitio apropiado para su almacenamiento, como centros de tratamiento, rellenos sanitarios o puntos de acopio intermedios (MITECO, 2024).

- Tratamiento final

Es la fase final del manejo de desechos, donde se procura eliminar o recuperar parte de ellos según su clasificación, ya sea para su tratamiento o disposición final.

Se implementan principalmente dos sistemas de recolección de residuos sólidos, adaptados a las particularidades geográficas y demográficas de la ciudad. A continuación, se detalla cada uno:

- **Recolección Pie de Vereda:** Sistema que implica la recolección de residuos depositados en fundas plásticas frente a los domicilios, mediante vehículos de carga posterior. Opera con frecuencias diurnas y nocturnas específicas para diferentes sectores de la ciudad, atendiendo aproximadamente a 3 millones de habitantes y recolectando alrededor de 2,000 toneladas diarias de residuos asimilables a domésticos a través de 234 rutas.
- **Recolección Contenerizada de Superficie:** Sistema basado en la disposición de residuos en contenedores de gran capacidad ubicados estratégicamente, los cuales son retirados por vehículos de carga lateral. Este método ofrece beneficios como la mejora del ornato, la salud pública y el medio ambiente,

la eliminación de microbasurales, la disponibilidad del servicio las 24 horas, la protección contra animales y plagas, la optimización de recursos y la minimización de riesgos laborales.

2.2.18. Dirección de gestión ambiental y Riesgos

La Dirección de Gestión Ambiental y Riesgos tiene como propósito conservar y manejar de forma sustentable los recursos naturales de la provincia, mediante la planificación, ejecución y control de políticas, programas, planes y proyectos ambientales. Su labor garantiza un ambiente sano y equilibrado para todos los habitantes, promoviendo además la participación ciudadana en la conservación del patrimonio natural, la calidad ambiental y la mitigación del cambio climático.

Dentro de sus competencias, la gestión adecuada de los residuos sólidos juega un rol fundamental, ya que su recolección eficiente contribuye a la prevención de la contaminación, la conservación de los suelos, la protección de fuentes hídricas y la reducción de riesgos ambientales. De esta forma, el manejo responsable de los desechos sólidos se integra como una acción clave para lograr un entorno más limpio, saludable y sostenible en la provincia (Prefectura del Carchi, 2024).

2.2.19. COOTAD

El Código Orgánico de Organización Territorial, Autonomía y Descentralización (COOTAD) es el cuerpo legal que establece la organización político-administrativa del Estado ecuatoriano en su territorio. Define el régimen de los gobiernos autónomos descentralizados (GAD) y los regímenes especiales, garantizando su autonomía política, administrativa y financiera. Además, desarrolla un modelo de descentralización progresiva y fija las competencias, fuentes de financiamiento y mecanismos para corregir desequilibrios territoriales.

De entre el contenido del Código Orgánico de Organización Territorial, Autonomía y Descentralización (COOTAD), se destacan varios aspectos relevantes en cuanto a la gestión de la recolección de desechos sólidos, los cuales refuerzan la importancia de la correcta administración ambiental por parte de los gobiernos autónomos descentralizados:

- La recolección, manejo y disposición de desechos sólidos es una competencia exclusiva de los gobiernos municipales, quienes deben garantizar un servicio adecuado para la protección ambiental y la salud pública.
- Los municipios deben implementar sistemas de gestión integral de residuos que incluyan medidas de prevención de la contaminación, promoción del reciclaje y reducción de residuos.
- Se permite a los municipios establecer tasas o tarifas mediante ordenanzas para financiar y sostener el servicio de recolección de desechos.
- Se promueve la participación comunitaria en la vigilancia y gestión de los servicios ambientales, fortaleciendo el control social y la corresponsabilidad ciudadana

La descentralización, impulsada por el Cootad, permite a los Gobiernos Autónomos Descentralizados (GAD) asumir competencias clave con autonomía, recursos y capacidades. En recolección de basura, esto significa que los municipios pueden gestionar directamente este servicio, adaptándolo a las necesidades locales, optimizando rutas, recursos y tecnologías, y fortaleciendo la eficiencia del servicio mediante decisiones más cercanas a la ciudadanía (Del Pozo, 2019).

III. METODOLOGÍA

3.1. ENFOQUE METODOLÓGICO

3.1.1. Enfoque

- **Cualitativo:** Se centra en comprender y describir las características, experiencias y significados de los fenómenos sociales. No se basa en datos numéricos, sino en atributos no cuantificables, como las percepciones o emociones humanas. Es un proceso flexible y circular, sin un orden estricto, lo que permite adaptarse a la complejidad del comportamiento humano y del contexto en estudio (Saramaniego, 2022).
- **Cuantitativo:** Se basa en el uso de métodos estadísticos y mediciones numéricas para analizar fenómenos. Utiliza herramientas como la observación estructurada, la medición de variables, el muestreo y el tratamiento estadístico. Su proceso es secuencial y riguroso, ya que sigue pasos definidos con el objetivo de comprobar hipótesis de manera precisa y objetiva (Saramaniego, 2022).

El presente proyecto adopta un enfoque de investigación mixto, combinando metodologías tanto cuantitativas como cualitativas. La metodología cuantitativa se aplica mediante el análisis de datos obtenidos en campo, como los tiempos y distancias de recolección, complementados con los resultados de encuestas estructuradas aplicadas a los ciudadanos de Tulcán, que evalúan aspectos como la frecuencia, puntualidad y eficiencia del servicio de recolección de desechos sólidos. Por su parte, la metodología cualitativa se manifiesta a través de la observación directa de las condiciones operativas del servicio, explorando factores prácticos que afectan la ejecución de las rutas. Esta integración permite comprender tanto el desempeño técnico del sistema como las condiciones reales del entorno, facilitando el diseño de rutas más eficientes y adaptadas al contexto local.

3.1.2. Tipo de Investigación

Se proporcionó una base teórica y técnica necesaria para desarrollar el aplicativo de rutas óptimas. Mediante una revisión sistemática de libros, revistas y artículos científicos, se identifican algoritmos de optimización y metodologías de desarrollo pertinentes. Este procedimiento facilitó la identificación de prácticas óptimas en el diseño de software para la recolección de desechos sólidos y la construcción de una base metodológica sólida para desarrollar el aplicativo.

3.1.2.1. Investigación Documental

Se centró en el análisis de sistemas similares y de la normativa relacionada con la gestión de desechos sólidos, lo cual permitió comprender los requisitos técnicos y operativos que debe cumplir el aplicativo. Esta investigación fue clave para acceder a datos históricos y estadísticas sobre sistemas de rutas existentes, lo que facilitó la identificación de problemas comunes y soluciones efectivas aplicadas en otros contextos.

3.1.2.2. Investigación de campo

Permitió recolectar datos directamente del entorno, incluyendo tanto la observación directa de las rutas actuales como entrevistas a conductores y personal operativo. Esta metodología facilitó la comprensión de las limitaciones prácticas del entorno y los factores operativos que afectan la eficiencia de las rutas. La interacción con los actores involucrados proporciona información valiosa sobre preferencias de usuario y requerimientos específicos que deben considerarse en el diseño del aplicativo, asegurando que el sistema final sea tanto eficiente.

3.2. IDEA A DEFENDER

La sistematización del sistema de gestión de rutas contribuirá para la recolección de desechos sólidos en la ciudad de Tulcán.

3.3. DEFINICIÓN Y OPERACIONALIZACIÓN DE LAS VARIABLES

3.3.1. Operacionalización variable independiente

Tabla 10. Variable independiente "Sistema de rutas"

Variable	Concepto	Dimensiones	Indicadores	Técnica	Instrumento
Sistema de gestión de rutas	Es una aplicación que optimiza el recorrido entre varios puntos geográficos, utilizando datos como tráfico, distancias y mapas para planificar rutas eficientes, reducir tiempos y mejorar la logística del transporte. (Flórez, 2023)	Planificación de rutas	Existencia de rutas definidas	Análisis documental	Cuestionario
		Cobertura geográfica	Zonas cubiertas por las rutas	Encuesta	
		Frecuencia de recorridos	Número de recorridos por zona	Entrevista	
		Tecnología aplicada	Registro digital	Entrevista	
		Coordinación operativa	Distribución de vehículos	Encuesta	

3.3.2. Operacionalización variable dependiente

Tabla 11. Variable Dependiente "Recolección desechos sólidos"

Variable	Concepto	Dimensiones	Indicadores	Técnica	Instrumento
Recolección de desechos sólidos	Corresponde al servicio que recoge y transporta residuos a centros de tratamiento. Dirigido a personas dentro del área de cobertura, mediante solicitud presencial ante la Alcaldía. (GADMCM, 2025)	Cobertura del servicio	Sectores o zonas atendidas	Encuesta Estructurada	Cuestionario
		Puntualidad	Cumplimiento del horario de recolección		
		Frecuencia de recorridos	Número de recorridos por zona		
		Servicio	Recolección de desechos sólidos		

3.4. MÉTODOS UTILIZADOS

3.4.1. Métodos

Método inductivo y deductivo

- Método inductivo

Es un proceso de razonamiento que parte de la observación de hechos específicos para llegar a conclusiones generales. Se basa en patrones identificados a través de la experimentación y la observación empírica (Narvaez, 2024).

- Método deductivo

Este método parte de principios o teorías generales para derivar conclusiones específicas. Se enfoca en probar hipótesis mediante la deducción lógica, verificando si las premisas conducen a una conclusión válida en situaciones concretas (Narvaez, 2022).

Para el desarrollo del proyecto de investigación, se emplean los métodos inductivo y deductivo con el fin de abordar de manera completa la optimización de las rutas de recolección. El método inductivo facilita la identificación de información específica a través de la observación y el análisis de datos reales sobre el comportamiento de las rutas, mientras que el método deductivo valida las soluciones propuestas mediante la aplicación de principios generales de optimización de rutas. Esta combinación garantiza un enfoque equilibrado, que integra la práctica basada en la experiencia y los fundamentos teóricos.

3.4.2. Técnicas utilizadas

3.4.2.1. Entrevista Semiestructurada

Se aplicó la técnica de la entrevista semiestructurada a diversos actores vinculados con el sistema de recolección de desechos sólidos en Tulcán, con el propósito de obtener información profunda sobre su funcionamiento y aceptación. Para ello, se utilizó una guía de preguntas abiertas y flexibles que permitió adaptar la conversación según el desarrollo de cada entrevista. Destaca la charla con el Ing. Hair Lima, responsable del área de gestión ambiental del municipio, la cual proporcionó datos relevantes sobre la operatividad del sistema y permitió explorar aspectos específicos para una comprensión más completa del contexto.

3.4.2.2. Encuesta

Se empleó la técnica de la encuesta para evaluar la aceptación del servicio de recolección de desechos sólidos entre los habitantes de Tulcán, especialmente en la zona comercial. Se utilizó como instrumento un cuestionario estructurado con preguntas cerradas, lo cual permitió cuantificar el nivel de satisfacción y percepción del servicio. Esta técnica facilitó la recopilación eficiente de datos representativos, brindando una visión clara sobre la opinión de la ciudadanía y contribuyendo al análisis estadístico del estudio.

3.5. ANÁLISIS ESTADÍSTICO

En Tulcán se llevó a cabo una encuesta sobre el servicio de gestión de desechos sólidos, obteniendo información clave acerca de la eficiencia y efectividad del sistema de manejo de desechos en la ciudad. Se abordaron temas como la frecuencia y calidad del servicio de recolección, el estado y la disponibilidad de los contenedores, y la accesibilidad a los puntos de reciclaje. Además, se evaluó la percepción de los ciudadanos respecto al servicio ofrecido. Los hallazgos revelaron áreas que requieren mejoras, proporcionando una base para tomar decisiones fundamentadas que optimicen la gestión de desechos sólidos y promuevan prácticas más sostenibles en la ciudad.

3.5.1. Muestreo probabilístico

Aleatorio simple:

Se aplicó el muestreo aleatorio simple para garantizar que cada habitante tuviera la misma probabilidad de ser seleccionado, lo cual mejora la representatividad y validez estadística de los resultados. Este método, eficiente y económico, fue ideal para el contexto urbano del estudio y permitió extrapolar los datos obtenidos a toda la población.

Formula:

$$n = \frac{Z^2 * p * q * N}{e^2(N - 1) + Z^2 * p * q} = 382$$

N = Población (63 972) Censo de los 2010 últimos datos recopilados

n = muestra = 382

p = probabilidad a favor (0,50)

q = probabilidad en contra (0,50)

z = nivel de confianza (95% → 1,96)

e = error de muestra (5% → 0,05)

IV. RESULTADOS Y DISCUSIÓN

4.1. RESULTADOS

4.1.1. Propuesta

Proponer una solución tecnológica para optimizar las rutas utilizadas en la recolección de desechos sólidos en la ciudad de Tulcán.

4.1.2. Resultados de la encuesta

Encuesta

El propósito de esta encuesta es recolectar información de los habitantes de Tulcán sobre el proceso de gestión de desechos sólidos, con el objetivo de analizar e interpretar los datos obtenidos.

Pregunta 1. ¿Conoce el horario de los camiones recolectores de basura que pasan por su sector?

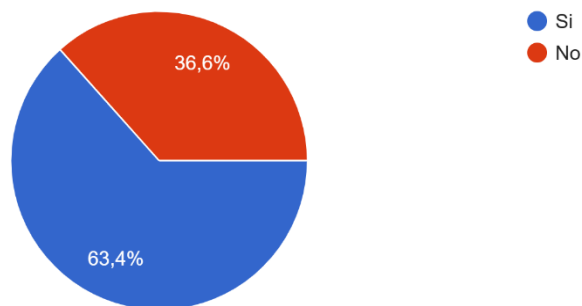


Figura 4. Horario de servicio de los camiones recolectores

Análisis e interpretación: Este dato evidencia la necesidad de mejorar los canales de comunicación del municipio, promoviendo el acceso a información clara y oportuna mediante estrategias digitales o impresas, con el fin de optimizar la participación ciudadana y garantizar una gestión eficiente del servicio de recolección de desechos.

Pregunta 2. ¿El camión recolector pasa a la hora esperada por su localidad?

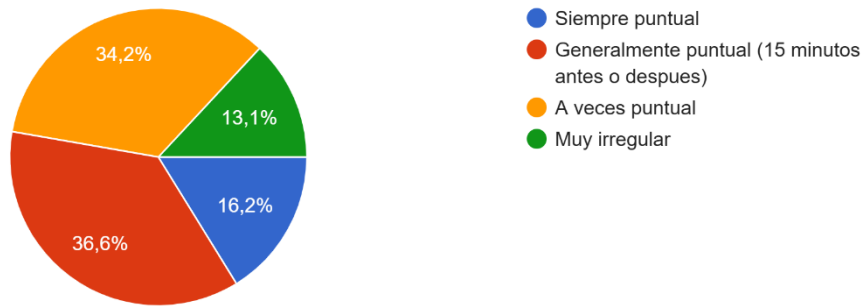


Figura 5. Horario de recolección

Análisis e interpretación: Estos datos reflejan una percepción generalizada de inestabilidad en el cumplimiento del servicio, lo que podría generar molestias en los usuarios y afectar la eficiencia del sistema. Es fundamental que las autoridades analicen las causas de esta variabilidad y evalúen estrategias para mejorar la puntualidad del servicio.

Pregunta 3. ¿Cuántos días en semana pasa el camión recolector por su sector?

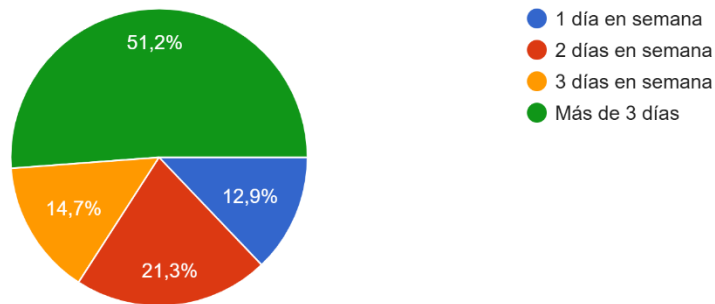


Figura 6. Días de recolección

Análisis e interpretación: El resultado muestra que la mayoría de los encuestados afirma que el camión recolector pasa más de tres días a la semana por su sector, lo cual evidencia una cobertura frecuente del servicio. No obstante, un porcentaje menor, aunque significativo, indica que el camión pasa entre uno y tres días por semana. Esta variabilidad sugiere diferencias en la planificación del servicio según el sector. Es necesario revisar la equidad en la frecuencia de recolección para asegurar que todas las zonas cuenten con un servicio suficiente y adecuado a sus necesidades.

Pregunta 4. ¿Ha observado que el camión recolector tenga que pasar varias veces por la misma calle?

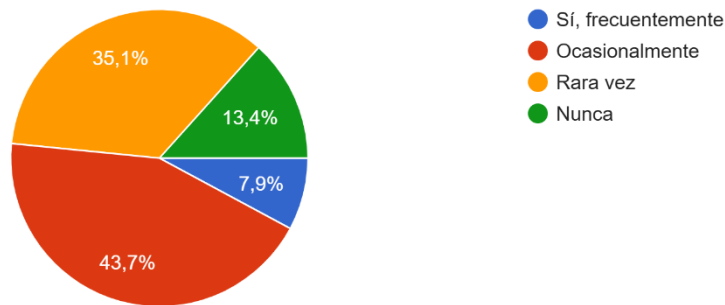


Figura 7. Frecuencia con la que pase el camión recolector

El resultado indica que una mayoría sustancial de los encuestados ha observado que el camión recolector pasa varias veces por la misma calle. Esto sugiere una posible ineficiencia en la planificación de rutas que requiere análisis para optimizar el servicio y reducir posibles inconvenientes.

Pregunta 5. El tiempo que tarda el camión en recolectar los residuos en su cuadra es:

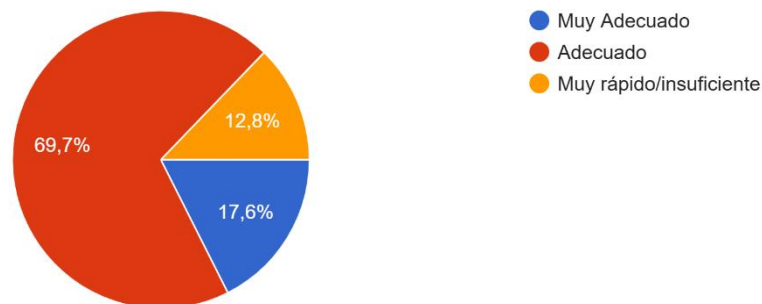


Figura 8. Tiempo de recolección

El resultado muestra que la mayoría de los encuestados califica como adecuado el tiempo que tarda el camión en recolectar los residuos en su cuadra. Una porción menor lo considera muy adecuado, mientras que una minoría percibe el tiempo como muy rápido o insuficiente. Esto indica una satisfacción general con la duración del servicio, aunque algunos usuarios experimentan que la recolección es demasiado breve.

Pregunta 6. ¿El horario actual de su ruta se ajusta a sus necesidades?

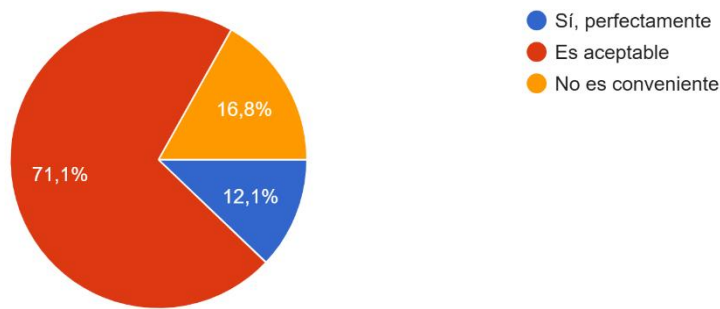


Figura 9. Conformidad con el horario actual

El resultado revela que una mayoría considerable de los encuestados considera que el horario actual de la ruta del camión recolector es aceptable. Una porción menor encuentra que se ajusta perfectamente a sus necesidades, mientras que una minoría significativa opina que no es conveniente. Esto sugiere que, si bien la mayoría tolera el horario actual, existe un grupo importante para el cual el horario no resulta ideal.

Pregunta 7. ¿Qué problemas ha identificado en su ruta de recolección?

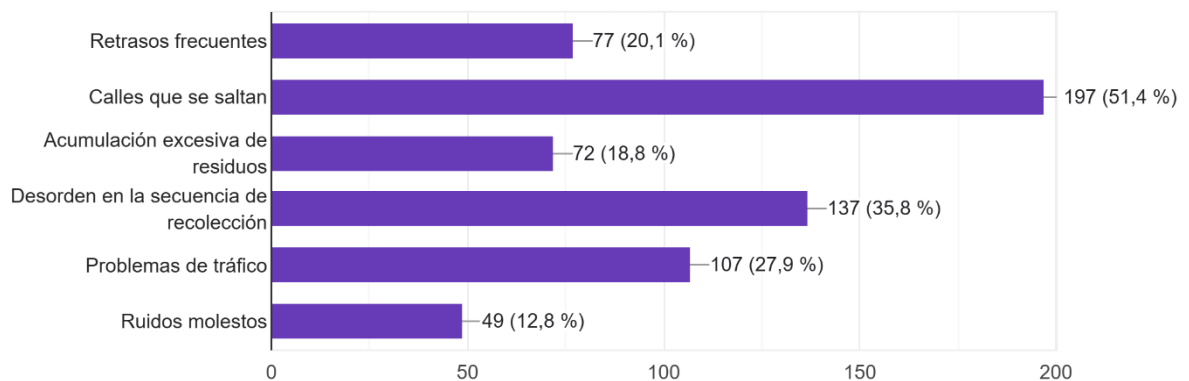


Figura 10. Identificación de problemas la ruta de recolección

El resultado muestra que el problema más identificado en la ruta de recolección es que se omiten calles. Otros problemas significativos incluyen el desorden en la secuencia de recolección, los problemas de tráfico y la acumulación excesiva de residuos. Los retrasos frecuentes y los ruidos molestos son los problemas menos identificados. Esto indica que la omisión de calles es una preocupación principal para los usuarios, sugiriendo posibles fallos en la cobertura de la ruta.

Pregunta 8. ¿Los contenedores/puntos de recolección asignados en su ruta son adecuados?

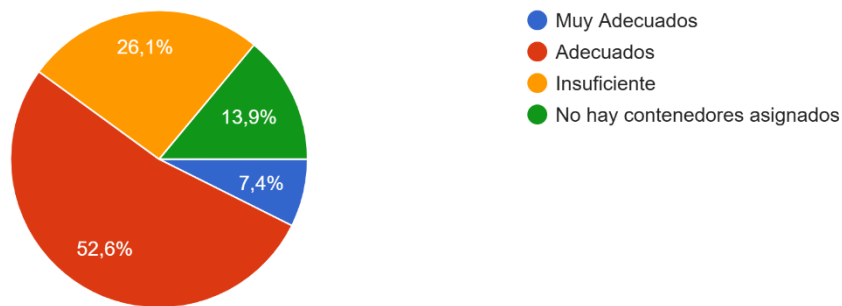


Figura 11. Puntos de recolección

El resultado indica que la mayoría de los encuestados considera que los contenedores o puntos de recolección asignados en su ruta son adecuados. Una porción significativa los percibe como muy adecuados, mientras que una minoría los encuentra insuficientes. Adicionalmente, una pequeña parte de los encuestados reporta no tener contenedores asignados. Esto sugiere una satisfacción general con la disposición de los puntos de recolección, aunque existen áreas donde la adecuación es cuestionable o donde directamente no se han asignado contenedores.

Pregunta 9. ¿A qué distancia está el punto de recolección más cercano?

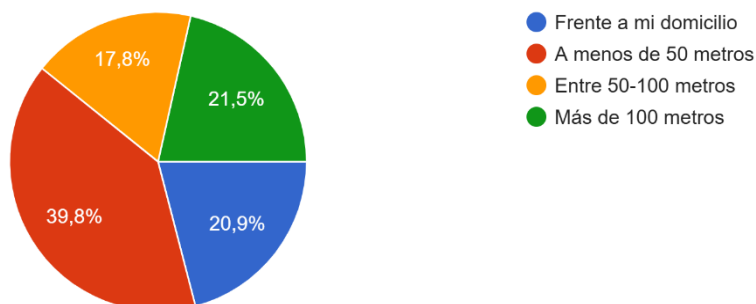


Figura 12. Distancia de puntos de recolección

El resultado muestra que una porción considerable de los encuestados tiene el punto de recolección más cercano frente a su domicilio o a menos de 50 metros de distancia. Un grupo similar reporta una distancia entre 50 y 100 metros, mientras que una parte menor indica que el punto de recolección más cercano está a más de 100 metros. Esto sugiere que, para una mayoría de la población, el acceso a los puntos de recolección es relativamente cercano.

Pregunta 10. ¿Ha notado si el camión se llena antes de completar la ruta?

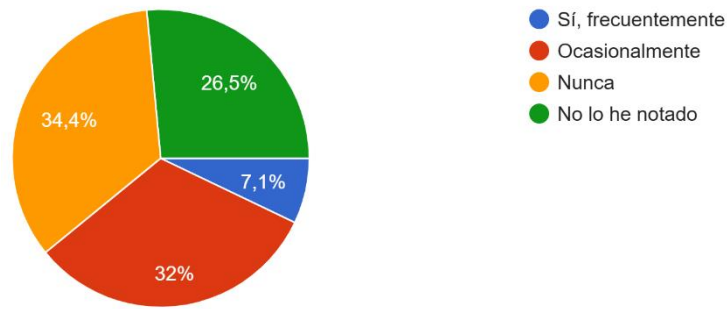


Figura 13. Exceso de desechos sólidos

El resultado indica que una parte significativa de los encuestados ha notado que el camión se llena antes de completar la ruta, ya sea frecuente u ocasionalmente. Una porción similar nunca lo ha notado, mientras que una minoría no ha prestado atención a esta situación. Esto sugiere que el llenado prematuro del camión es una ocurrencia observable para una parte importante de la población, lo que podría implicar ineficiencias en la capacidad del vehículo o en la planificación de la ruta.

Pregunta 11. ¿Cómo calificaría la velocidad del servicio de recolección?

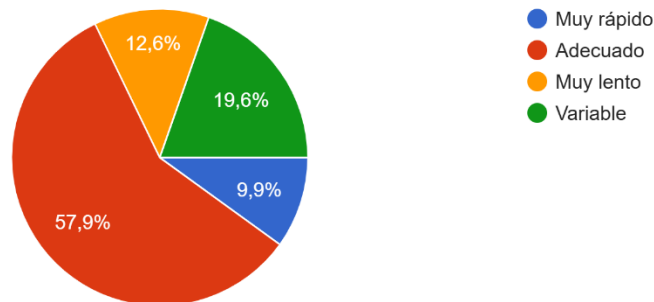


Figura 14. Velocidad del servicio de recolección

El resultado muestra que la mayoría de los encuestados califica la velocidad del servicio de recolección como adecuada. Una porción considerable la percibe como variable, mientras que minorías la consideran muy rápida o lenta. Esto sugiere una satisfacción general con la velocidad del servicio, aunque existe una variabilidad notable que afecta a una parte importante de los usuarios.

Pregunta 12. ¿Ha notado semanas o días que el servicio de recolección no ha cumplido con su servicio?

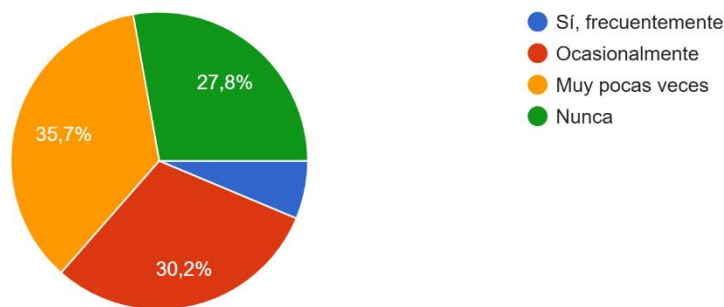


Figura 15. Servicio de recolección

El resultado indica que una proporción importante de los encuestados ha notado semanas o días en que el servicio de recolección no ha cumplido con su servicio, ya sea ocasionalmente o muy pocas veces. Una porción similar afirma que nunca ha notado esta situación, mientras que una minoría lo ha experimentado frecuentemente. Esto sugiere que la inconsistencia en el cumplimiento del servicio es una realidad para una parte considerable de los usuarios.

Pregunta 13. ¿Existen condiciones especiales en su sector que afecten la recolección?

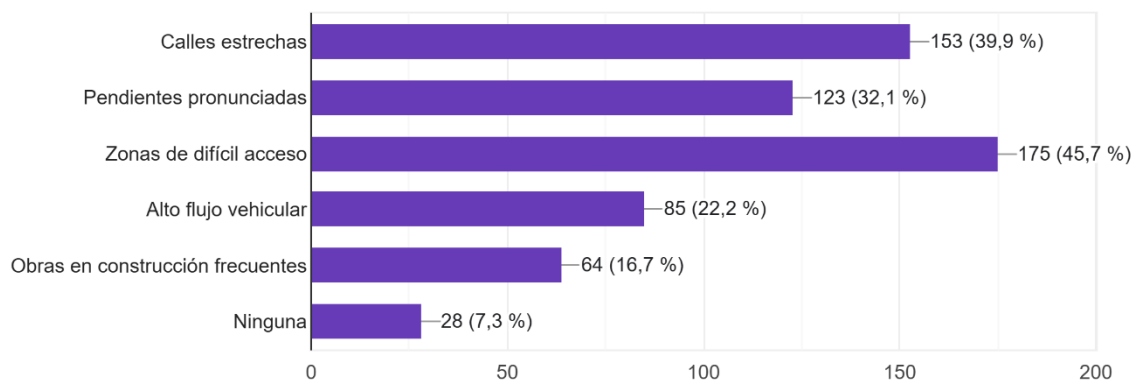


Figura 16. Acumulación de desechos sólidos

El resultado revela que una proporción significativa de los encuestados identifica zonas de difícil acceso como una condición especial que afecta la recolección en su sector. Otros factores relevantes incluyen calles estrechas y pendientes pronunciadas. El alto flujo vehicular y las obras en construcción frecuentes son mencionados en menor medida, mientras que solo una pequeña parte de los encuestados indica que no existen condiciones especiales que afecten la recolección. Esto sugiere que las características geográficas y de infraestructura de algunos sectores representan desafíos importantes para la eficiencia del servicio de recolección.

Pregunta 14. ¿En qué días de la semana observa mayor acumulación de residuos?

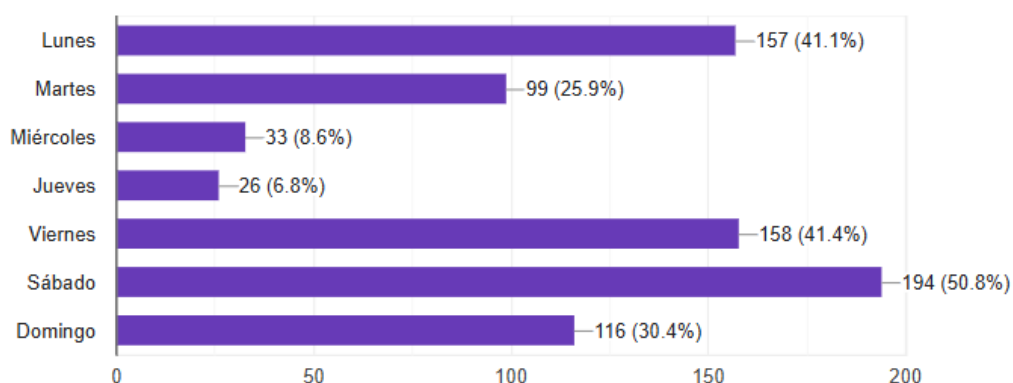


Figura 17. Días de mayor acumulación de desechos sólidos

El resultado muestra que la mayor acumulación de residuos se observa los sábados, seguido de cerca por los viernes y los lunes. El domingo también presenta una acumulación considerable. En contraste, los miércoles y jueves son los días en que se percibe menor acumulación de residuos. Esto sugiere una pauta semanal en la generación de residuos, con un pico hacia el final de la semana y al inicio de la siguiente.

Pregunta 15. ¿Qué cambios sugiere para mejorar la eficiencia de su ruta de recolección?

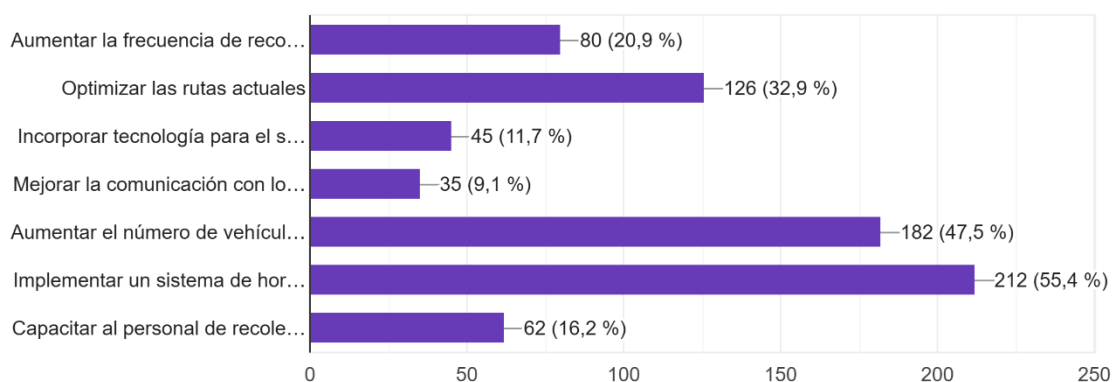


Figura 18. Sugerencia de cambios

El resultado muestra que la mayor acumulación de residuos se observa los sábados, seguido de cerca por los viernes y los lunes. El domingo también presenta una acumulación considerable. En contraste, los miércoles y jueves son los días en que se percibe menor acumulación de residuos. Esto sugiere una pauta semanal en la generación de residuos, con un pico hacia el final de la semana y al inicio de la siguiente.

4.1.3. Resultado de la entrevista

La entrevista se aplicó al encargado del departamento de gestión ambiental del municipio de Tulcán, quien tiene a su cargo la supervisión y funcionamiento del sistema de recolección de desechos sólidos. A través de esta entrevista se buscó recopilar información clave sobre los procesos actuales, desafíos y limitaciones del sistema, con el objetivo de identificar los factores más relevantes que permitan fundamentar y orientar adecuadamente la propuesta final del proyecto de tesis, enfocada en el diseño de rutas óptimas para una gestión más eficiente y sostenible de los residuos en la ciudad.

Tabla 12. Resultado de la Entrevista

Nº	Pregunta	Respuesta
1	¿Cuánto tiempo se requiere para completar la recolección total de basura en la ciudad?	En Tulcán, la recolección de desechos se realiza diariamente en las zonas urbanas y de forma interdiaria en las zonas rurales. El servicio opera durante 7 horas, de 6 a.m. a 1 p.m. En las parroquias rurales, la recolección se efectúa entre 1 y 2 veces por semana.
2	¿Cuál es el tamaño del departamento de ambiente (número de empleados, ingresos anuales, etc.)?	El servicio de recolección inicia con la reunión del personal en un punto de encuentro. Se dispone de 7 vehículos: 5 carritos recolectores, 1 camión plataforma y 1 camión para reciclaje. Cada carrito cuenta con un chofer y 2 auxiliares asignados a una zona específica durante un mes.
3	¿Cuál es la estructura organizativa y los departamentos relevantes para el proyecto?	Con los departamentos que se tiene convenios son Jefatura de Tránsito.
4	¿Cuál es la definición del ámbito de manejo de desechos sólidos?	En la mayoría de las ciudades de Ecuador, el GAD municipal tiene la responsabilidad de garantizar la limpieza de la ciudad, lo que lo convierte en el principal encargado del manejo y tratamiento de los desechos sólidos.
5	¿Cuáles son las tendencias y desafíos actuales en el sector?	El departamento de gestión ambiental enfrenta desafíos económicos y técnicos, entre ellos un personal mayor de 50 años que afecta la eficiencia del servicio y camiones en mal estado. La tarifa actual por el servicio es baja, con un pago mensual de 0.28 centavos, tras la reducción del subsidio municipal del 60% al 44%. Aunque la cobertura en la ciudad alcanza el 95%, en las zonas rurales se reduce al 60%, lo que genera insatisfacción entre los usuarios.
6	¿Cuáles son las normativas y regulaciones relevantes?	El servicio de recolección de basura tiene un costo aproximado de 2 millones de dólares, con un subsidio del 44%. Se espera reducir este gasto con el tiempo. El área opera bajo normativas como las ordenanzas municipales, la Ley Orgánica de Economía Circular y regulaciones del GAD municipal.
7	¿Cuál es la descripción del sistema actual para la recolección de desechos sólidos?	En la ciudad se utiliza un sistema de contenedores para la recolección de desechos sólidos, permitiendo a los ciudadanos depositarlos en puntos designados. En zonas sin contenedores, se aplica el método "A pie y vereda", donde los recolectores recogen directamente desde las veredas. Las zonas de recolección están organizadas de la siguiente manera: Rumichaca – Argentina con recolección a pie y vereda, Argentina – Boyacá, Boyacá – Unión 2 calles, Unión 2 calles –

		Andrés Bello, y Andrés Bello – Obelisco, todas con recolección contenerizada, mientras que las parroquias de Julio Andrade, Santa Marta, Carmelo y otras no especifican el tipo de recolección. Actualmente, operan siete vehículos de recolección, cada uno con un conductor y dos auxiliares. Cada equipo se dedica a una zona de recolección durante un mes, rotando al inicio de cada nuevo mes, lo que permite cubrir las seis zonas a lo largo del año.
8	¿Cuáles son los métodos empleados para planificar las rutas de recolección?	La recolección sigue una planificación mensual. A cada conductor, identificado por un código como el 0001, se le asignan dos auxiliares para trabajar en una zona específica durante todo el mes. El servicio se realiza en la mañana para evitar congestión vehicular, aunque en casos especiales, como fallas mecánicas, se traslada a la noche.
9	¿Cuáles son las tecnologías o sistemas existentes para la gestión de rutas?	La gestión del servicio se basa en métodos empíricos, sin un enfoque formal. Se utiliza un sistema GPS en los vehículos recolectores para obtener datos sobre rutas y duración, generando un mapa cada dos meses. Además, anualmente se evalúa el peso de la basura recolectada para identificar las rutas con mayor contaminación.
10	¿Cuáles son los datos disponibles sobre los desechos sólidos (volumen, tipo, ubicación, etc.)?	Desde 2019 se han implementado actividades como la caracterización, pesaje y cálculo del volumen de desechos sólidos, así como el monitoreo de su generación por habitante. Inicialmente, se producían 0.63 kg por persona al día, cifra que ha aumentado a 0.72 kg diarios en la actualidad.
11	¿Cuáles son los datos históricos de recolección de desechos sólidos (rutas, tiempos, eficiencia, etc.)?	En la ciudad de Tulcán se recogen cada día aproximadamente 64.5 toneladas de desechos sólidos, de las cuales cerca de 7.37 toneladas tienen potencial de reciclaje. A nivel urbano en Ecuador, el 55% de los residuos corresponde a materia orgánica, mientras que el 45% restante es de tipo inorgánico.
12	¿Qué sistemas de información geográfica (SIG) o software son empleados para la planificación de los recorridos?	No cuentan, pero se ha realizado un seguimiento de las rutas con una app
13	¿Cuáles son las herramientas de gestión y monitoreo de rutas existentes?	No existe ningún, verificación de ruta por GPS
14	¿Cuáles son los dispositivos de seguimiento y comunicación utilizados en los vehículos?	No tiene, la comunicación es vía celular
15	¿Cuáles son las limitaciones técnicas, operativas o financieras actuales?	La mayor limitación que se tiene es con respecto a lo financiero, debido a que los carros recolectores tienden a dañarse con el cual es un gasto considerable. En cuanto al parque automotor destinado a la recolección de desechos sólidos, se cuenta con un total de siete vehículos. De estos, dos datan del año 2014, uno fue adquirido en 2021 y los cuatro restantes son considerados vehículos antiguos.
16	¿Cuáles son las restricciones geográficas o de infraestructura que puedan afectar la planificación de rutas?	En las zonas rurales, la distancia limita la eficiencia del servicio, con recorridos de hasta 80 km, lo que aumenta los costos. En la ciudad, el máximo recorrido es de 40 km.
17	¿Cuáles son los factores ambientales relevantes?	Los factores que influyen son los días de invierno debido a que la recolección se complica por dispersión de la basura

18	¿Cuáles son los principales problemas ambientales asociados a la recolección y manejo de desechos sólidos?	Un desafío considerable se encuentra en la aparición de zonas susceptibles a infecciones, lo que representa un peligro para la salud de los habitantes.
19	¿Cuáles son las tendencias en la gestión de desechos sólidos?	Leyes propuestas por el país, reciclaje todo esto enfocado en la ley circular
20	¿Qué estrategias son más efectivas para el manejo adecuado de los desechos sólidos?	Fomentar la concienciación ciudadana para evitar la contaminación y promover prácticas responsables en la comunidad.
21	¿Cuáles son los factores que se consideran al planificar las rutas de recolección?	Que las personas tengan un lugar bueno para depositar su basura y sea accesible para los carros y personal de recolección.
22	¿De qué manera se evalúa la efectividad en la recolección de desechos sólidos?	Se realiza el pesaje de los desechos sólidos recolectados para determinar la cantidad exacta, con el propósito de analizar la eficiencia del proceso de recolección e identificar posibles inconvenientes.
23	¿Cuáles son los principales desafíos en la recopilación de datos sobre desechos sólidos?	La dificultad radica en que el cumplimiento de la ruta, el pesaje y la eficiencia del servicio no se abordan diariamente, sino que simplemente se consideran como completados.
24	¿Cuáles son las principales limitaciones de las tecnologías y herramientas existentes?	Una limitación está ligada con la falta de presupuesto para adquirir herramientas tecnológicas.
25	¿Qué innovaciones y avances tecnológicos están marcando tendencia en el desarrollo de herramientas para la gestión de desechos sólidos?	Se ha propuesto tecnología para los carros chip de seguimiento, pero por temas económicos no se ha podido implementar
26	¿Cuáles son los principales desafíos y limitaciones para implementar un sistema de rutas óptimas?	Se busca mejorar integralmente el sistema para establecer una empresa de servicios de limpieza, aunque la viabilidad es cuestionada por las restricciones económicas. A pesar de esto, se observa un progreso constante en la calidad del servicio, gracias a los pagos del municipio que favorecen la mejora continua.
27	¿Cómo cree que se pueden superar estos desafíos y limitaciones?	Voluntad política, concientización de la ciudadanía.

4.1.4. Fase de Planificación de requerimientos

En la reunión inicial con el Departamento de Ambiente se recopiló información clave sobre el proceso de recolección de desechos sólidos, lo que permitió definir los objetivos del sistema propuesto y sus requerimientos. Se priorizaron aspectos como funcionalidad, adaptabilidad y facilidad de uso, estableciendo una base sólida para el desarrollo del aplicativo. Además, se identificaron detalles del proceso actual en Tulcán, como horarios, personal, vehículos y zonas de recolección, lo que permitió

detectar falencias y proponer una solución tecnológica para optimizar la gestión de los residuos.

4.1.4.1. Requerimientos Funcionales

A continuación, se presentan los requerimientos funcionales y no funcionales, definidos con base en las características establecidas por la norma ISO/IEC 25000, la cual se fundamenta en la ISO/IEC 9126 sobre evaluación de la calidad de productos de software

Se mantuvo contacto con Msc. Hair Lima jefe del departamento de ambiente para el levantamiento de requerimientos.

Requerimientos funcionales

Tabla 13. Requerimientos funcionales

Requerimientos Funcionales	
ID requerimiento	RF001
Nombre	Visualizar mapa
Prioridad	Alta
Descripción	El sistema debe permitir al usuario visualizar un mapa interactivo que muestre una ubicación correspondiente a una zona
Entrada	Datos de la ruta asignada desde la base de datos
Proceso	Obtener datos de ruta desde el sistema y renderizar el mapa con la posición actual y el recorrido.
Salida	Mapa interactivo mostrado en la pantalla con marcadores de ubicación y ruta.
ID requerimiento	RF002
Nombre	Roles
Prioridad	Alta
Descripción	El sistema debe permitir el ingreso al aplicativo diferentes niveles de acceso y funciones, según el rol del usuario (administrador, chofer, auxiliar).
Entrada	Credenciales del usuario (para identificar el rol).
Proceso	Verificar el rol del usuario al iniciar sesión.
Salida	Aplicativo personalizado según el rol, con funciones y datos correspondientes.
ID requerimiento	RF003
Nombre	Visualizar información
Prioridad	Medía
Descripción	El sistema debe permitir a los usuarios visualizar información relevante según su rol, como datos de rutas, unidades de recolección, historial de recorridos y estado de tareas asignadas.
Entrada	Solicitud del usuario para acceder a una sección del sistema (por ejemplo: personal, vehículos, horarios y ciudad).
Proceso	Consultar la base de datos para obtener la información correspondiente.
Salida	Visualización de información textual, gráfica o tabular (por ejemplo: lista de personal, horarios, vehículos, horarios y ciudad).
ID requerimiento	RF004
Nombre	Gestión de base de datos (CRUD)
Prioridad	Alta

Descripción	El sistema debe permitir a los usuarios autorizados realizar operaciones de creación, lectura, actualización y eliminación (CRUD) de los registros almacenados en la base de datos, como usuarios, rutas, vehículos y horario.
Entrada	Datos ingresados por el usuario mediante formularios o interfaces del sistema Crear: Ingresar nuevos datos (por ejemplo, registrar un nuevo usuario).
Proceso	Leer: Consultar y visualizar registros existentes. Actualizar: Modificar información previamente almacenada. Eliminar: Borrar registros cuando ya no sean necesarios.
Salida	Confirmación de éxito o error de la operación realizada
ID requerimiento	RF005
Nombre	Generación de rutas
Prioridad	Alta
Descripción	El sistema debe permitir la generación automática o manual de rutas optimizadas para los vehículos de recolección, considerando factores como puntos de recolección, horarios y disponibilidad de unidades. Datos de puntos de recolección (ubicaciones geográficas, prioridad, frecuencia). Horarios disponibles.
Entrada	Estado de los vehículos. Parámetros definidos por el usuario (por ejemplo: tipo de ruta, zonas, tiempo estimado).
Proceso	Validar los datos de entrada
Salida	Rutas generadas visualizadas en el mapa
ID requerimiento	RF006
Nombre	Segmentación de zonas
Prioridad	Alta
Descripción	El sistema debe permitir la segmentación geográfica del área de cobertura en zonas específicas, con el fin de organizar las rutas y asignar personal.
Entrada	Límite geográfico por municipio o sector
Proceso	Almacenar la información de cada zona con un identificador único
Salida	Visualización de zonas segmentadas en el mapa
ID requerimiento	RF007
Nombre	Autenticación de usuarios
Prioridad	Alta
Descripción	El sistema debe permitir que los usuarios accedan mediante un proceso de autenticación, ingresando su nombre de usuario (o cédula) y contraseña. Solo los usuarios registrados y con credenciales válidas podrán acceder a las funcionalidades del sistema.
Entrada	Credenciales
Proceso	Verificar que el nombre de usuario exista en la base de datos.
Salida	Acceso al sistema si las credenciales son correctas

4.1.4.2. Requerimientos No funcionales

Tabla 14. Requerimientos no funcionales

Requerimientos No funcionales	
ID requerimiento	RNF001
Nombre	Disponibilidad
Prioridad	Alta
Descripción	El sistema debe garantizar una disponibilidad mínima del 99% durante el horario operativo establecido, asegurando que los usuarios puedan acceder a las

	funcionalidades críticas como visualización de rutas, autenticación y registro de recolecciones, sin interrupciones prolongadas.
Entrada	Solicitudes de acceso de usuarios
Proceso	Verificación del estado del servidor y disponibilidad del sistema
Salida	Acceso disponible o mensaje de mantenimiento
ID requerimiento	RNF002
Nombre	Usabilidad
Prioridad	Alta
Descripción	El sistema debe contar con una interfaz intuitiva y amigable que facilite su uso por parte de todos los roles del sistema (administradores, choferes, auxiliares), permitiendo la ejecución de tareas básicas sin necesidad de conocimientos técnicos avanzados.
Entrada	Interacción del usuario con la interfaz
Procesos	Renderizado de menús, botones e iconos claros
Salida	Navegación fluida e intuitiva
ID requerimiento	RNF003
Nombre	Rendimiento
Prioridad	Alta
Descripción	El sistema debe responder de forma eficiente ante las acciones del usuario, garantizando un tiempo de respuesta adecuado para las funciones críticas como autenticación, generación de rutas, carga de mapas y consultas a la base de datos.
Entrada	Solicitud de rutas desde la app
Procesos	Consulta optimizada a la base de datos y renderización
Salida	Lista de rutas en pantalla
ID requerimiento	RNF004
Nombre	Seguridad
Prioridad	Alta
Descripción	El sistema debe garantizar la protección de la información almacenada y procesada, implementando mecanismos de autenticación, autorización, cifrado y control de acceso para prevenir accesos no autorizados y asegurar la integridad y confidencialidad de los datos.
Entrada	Credenciales, datos personales
Procesos	Cifrado de contraseña
Salida	Datos protegidos
ID requerimiento	RNF005
Nombre	Compatibilidad
Prioridad	Alta
Descripción	El sistema debe ser compatible con los navegadores más utilizados y dispositivos móviles para asegurar su accesibilidad desde diferentes plataformas y entornos de usuario. Además, debe garantizar que las funcionalidades del sistema se mantengan intactas en diversas versiones de software.
Entrada	Dispositivo portátil
Procesos	Detección de dispositivo y aplicación de diseño responsivo
Salida	Visualización correcta y sin errores de formato

4.1.5. Historias de usuario

Tabla 15. Historia de usuario N°1

Historia de usuario	
Número	1
Rol	Administrador
Nombre historia	Planificación de rutas
Prioridad	Alta
Funcionalidad	Registrar y gestionar las rutas de recolección por zonas
Criterio de aceptación	Cobertura y recolección total de las zonas

Tabla 16. Historia de usuario N°2

Historia de usuario	
Número	2
Rol	Chofer
Nombre historia	Recorrido de las rutas
Prioridad	Alta
Funcionalidad	Conducción del vehículo recolector Supervisión de la recolección de desechos sólidos Descarga total de los desechos sólidos en el relleno sanitario
Criterio de aceptación	Cobertura de la zona

Tabla 17. Historia de usuario N°3

Historia de usuario	
Número	3
Rol	Auxiliar
Nombre historia	Ayudante de recolección
Prioridad	Alta
Funcionalidad	Recolección de desechos sólidos en las zonas de la ciudad
Criterio de aceptación	Cobertura de la zona

4.1.6. Fase de diseño de usuario

El coordinador del Departamento de Ambiente, durante la última semana de cada mes, una reunión para planificar las actividades del mes siguiente. En esta reunión se abordan temas importantes, como la revisión de notificaciones, llamados de atención y la evaluación del trabajo realizado durante el mes que está por concluir.

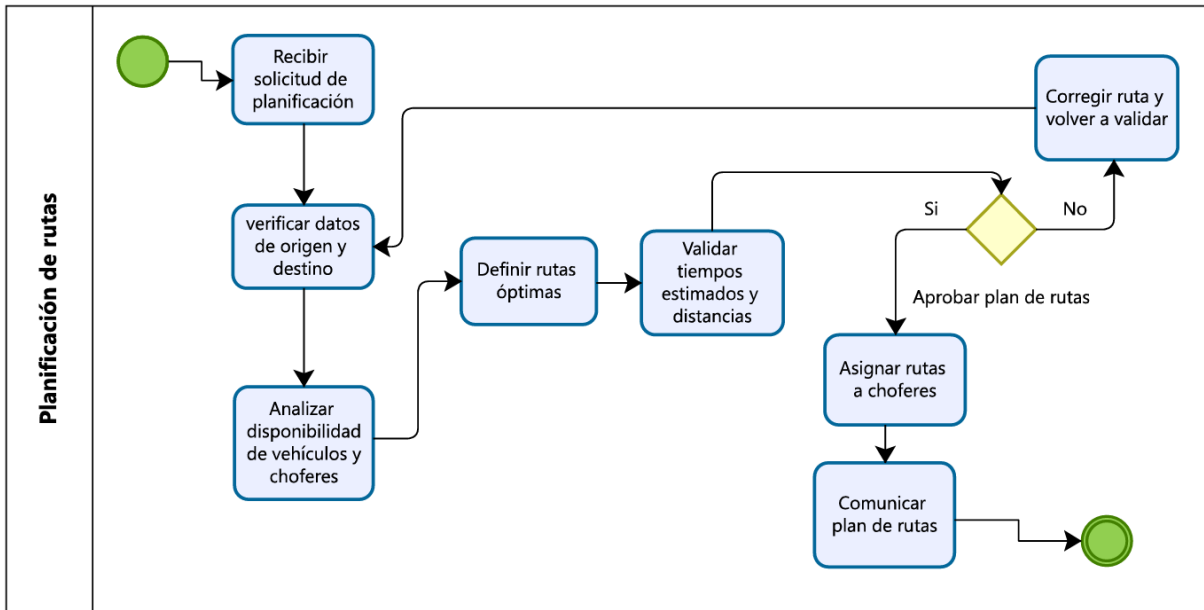


Figura 19. Gestión de rutas

La recolección de desechos sólidos en Tulcán se organiza por zonas, contando actualmente con cinco áreas definidas. Las actividades inician diariamente en el horario de 6:00 a.m. a 2:00 p.m. Todos los trabajadores se reúnen en su punto de encuentro, ubicado en la avenida Colón y 10 de agosto. Desde allí, cada equipo se dirige a su zona asignada.

La operación se realiza con un carro recolector, conformado por un chofer y dos auxiliares, quienes se encargan de recolectar la basura de toda la zona. Una vez finalizada la recolección, los equipos trasladan los desechos al relleno sanitario para su disposición final. Posteriormente, regresan al punto de encuentro inicial para concluir su jornada.

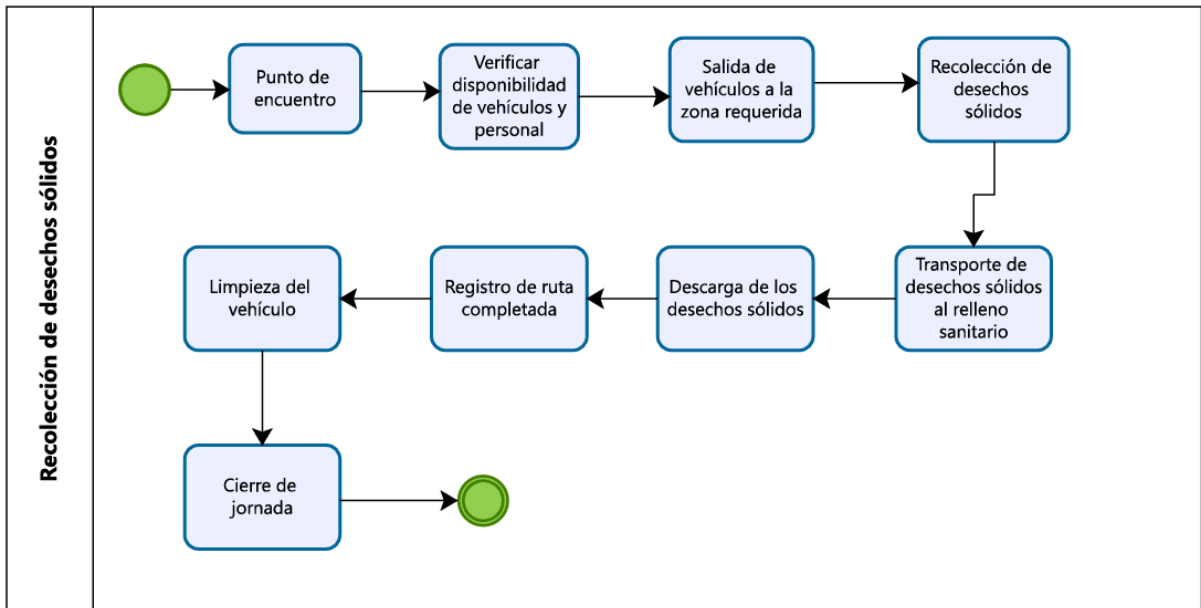


Figura 20. Proceso de recolección de desechos sólidos

4.1.7. Diagramas de caso de uso

En esta sección se presentan los casos de uso del sistema de gestión de rutas para la recolección de desechos sólidos. Estas representaciones son fundamentales para el análisis y diseño del software, ya que describen cómo interactúan los distintos usuarios, como el administrador, el chofer y el auxiliar, con el sistema. Los casos de uso permiten modelar las funcionalidades desde la perspectiva del usuario, asegurando que el producto final responda a sus necesidades y se alinee con sus expectativas.

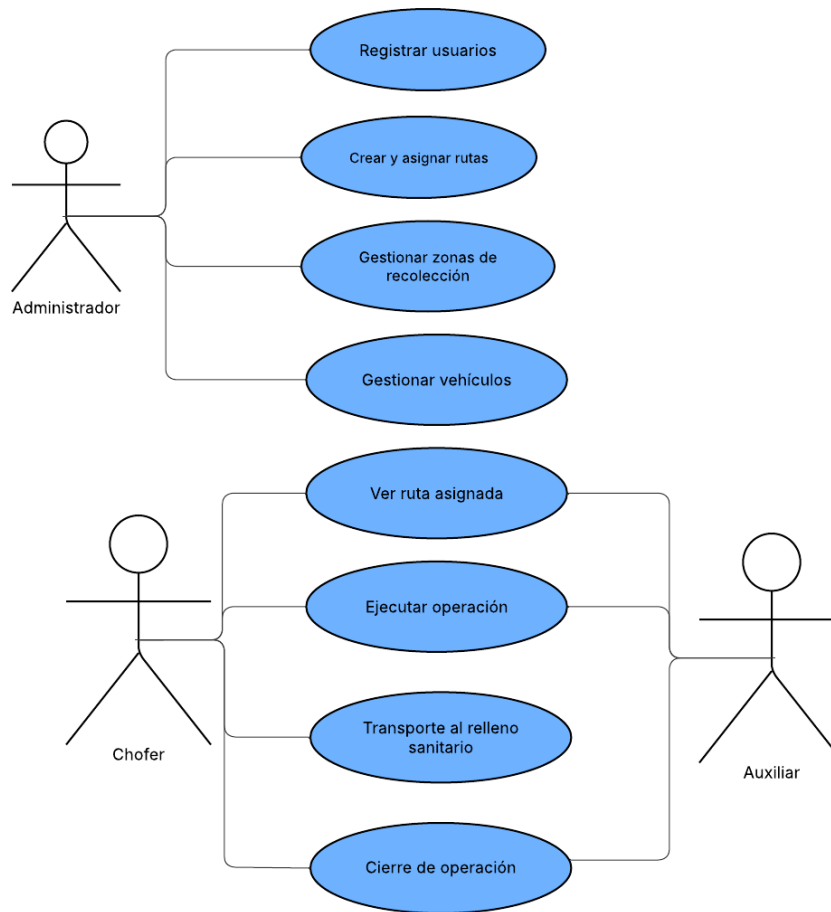


Figura 21. Caso de uso recolección de desechos

4.1.8. Diseño de la base de Datos

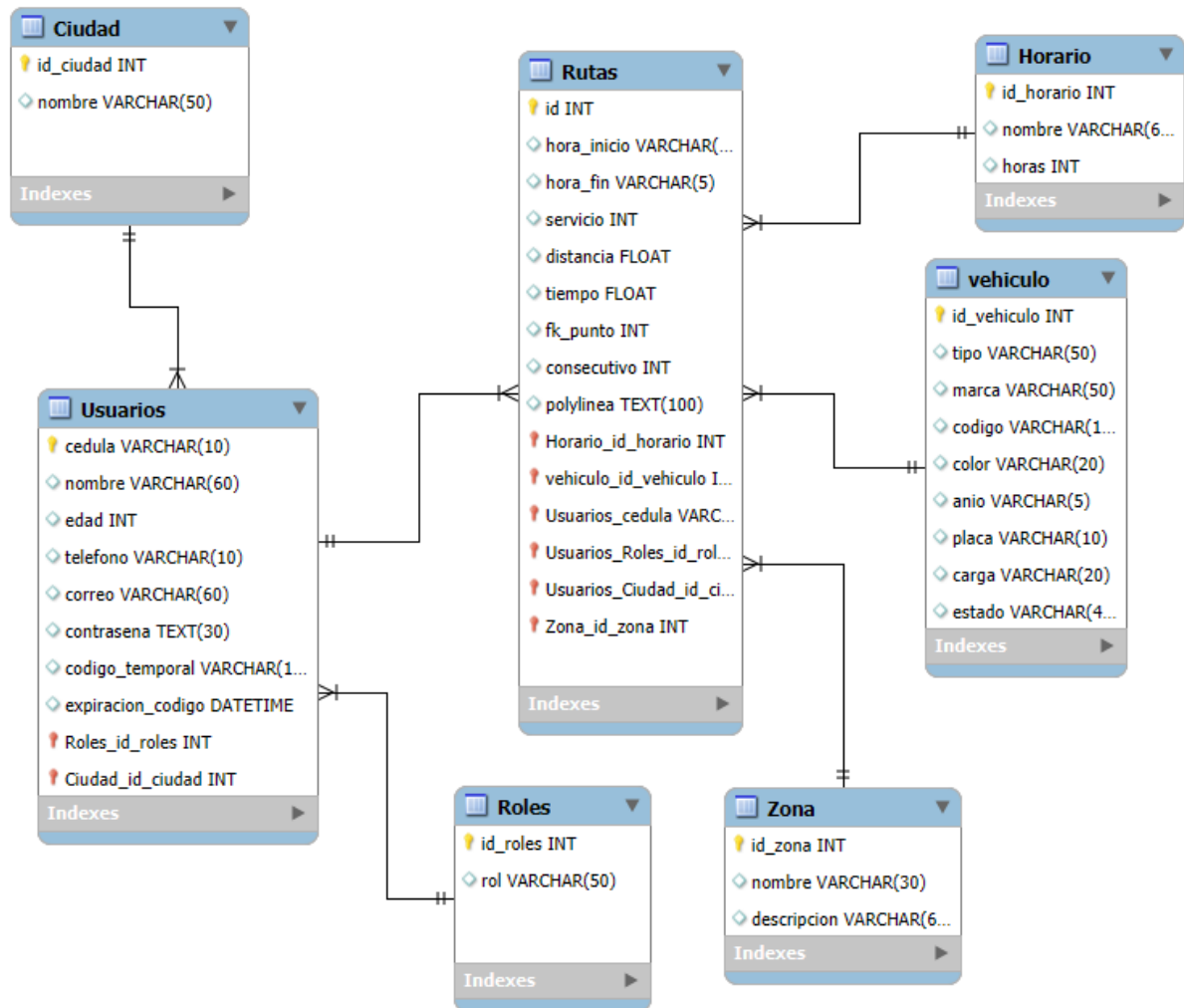


Figura 22. Diseño de la base de datos

4.1.9. Diseño de Interfaces

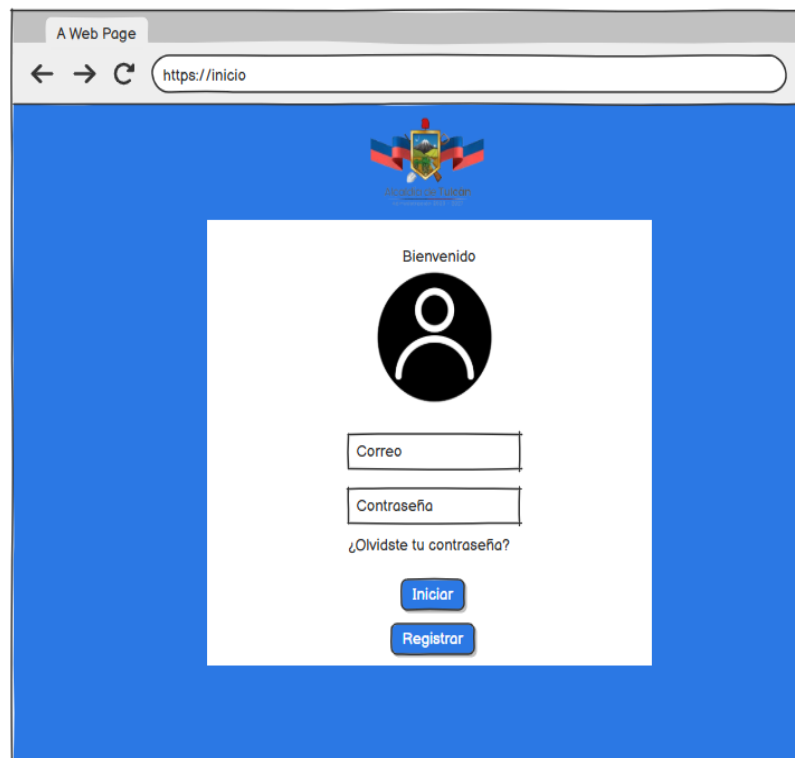


Figura 23. Prototipo pantalla de inicio

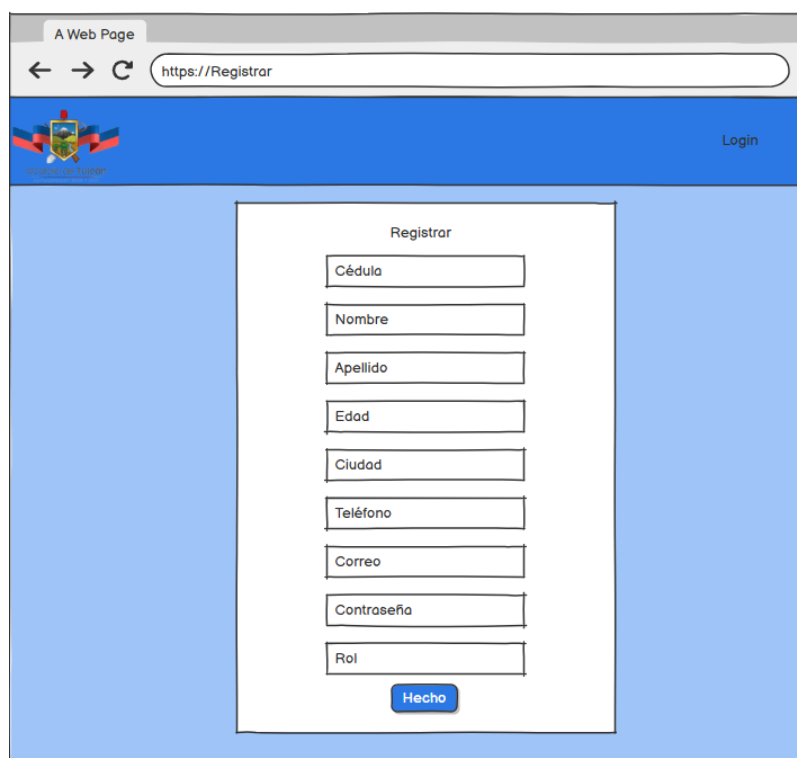


Figura 24. Prototipo pantalla registro de usuario

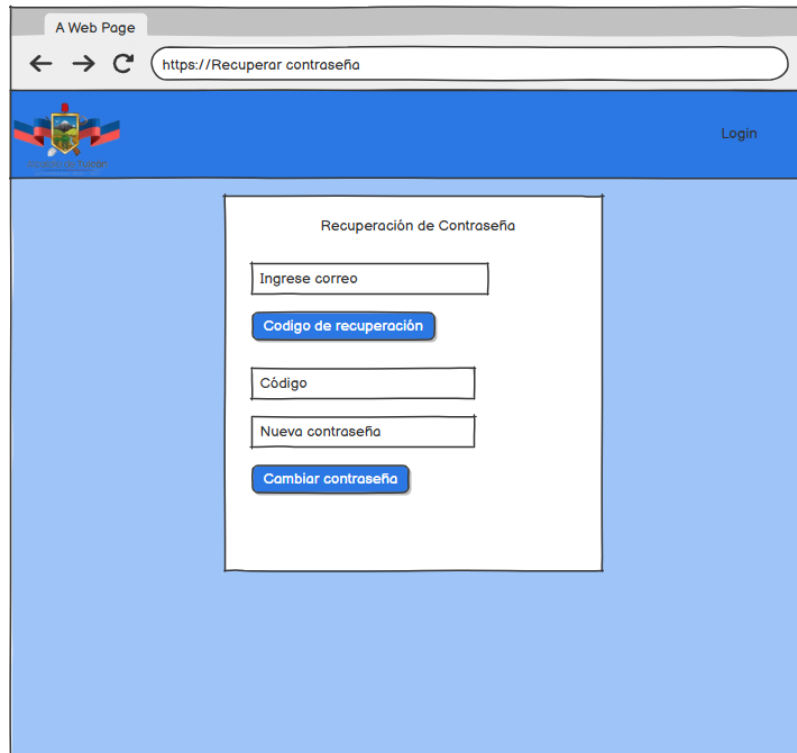


Figura 25. Prototipo pantalla de recuperación contraseña

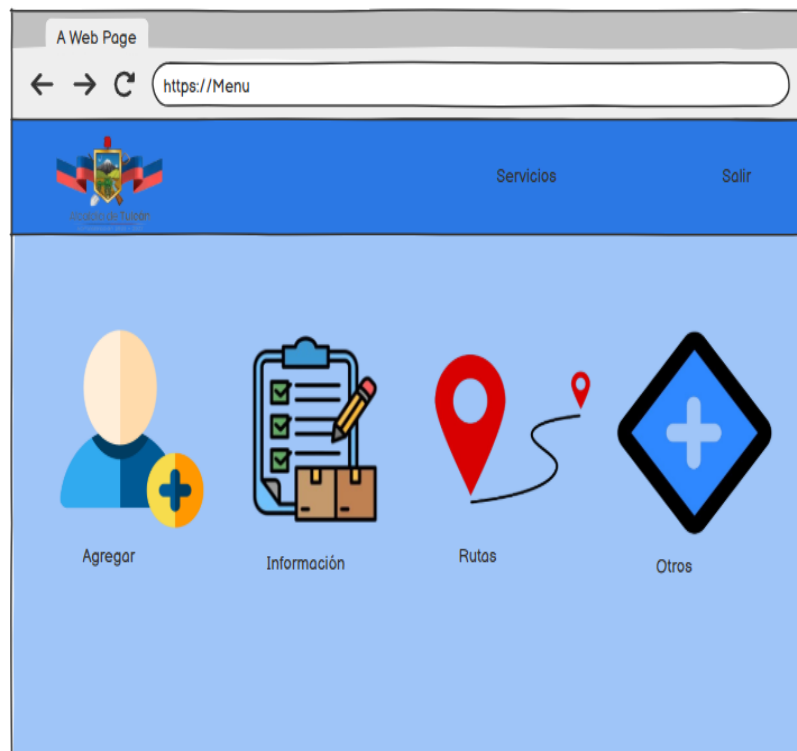


Figura 26. Prototipo pantalla menú principal

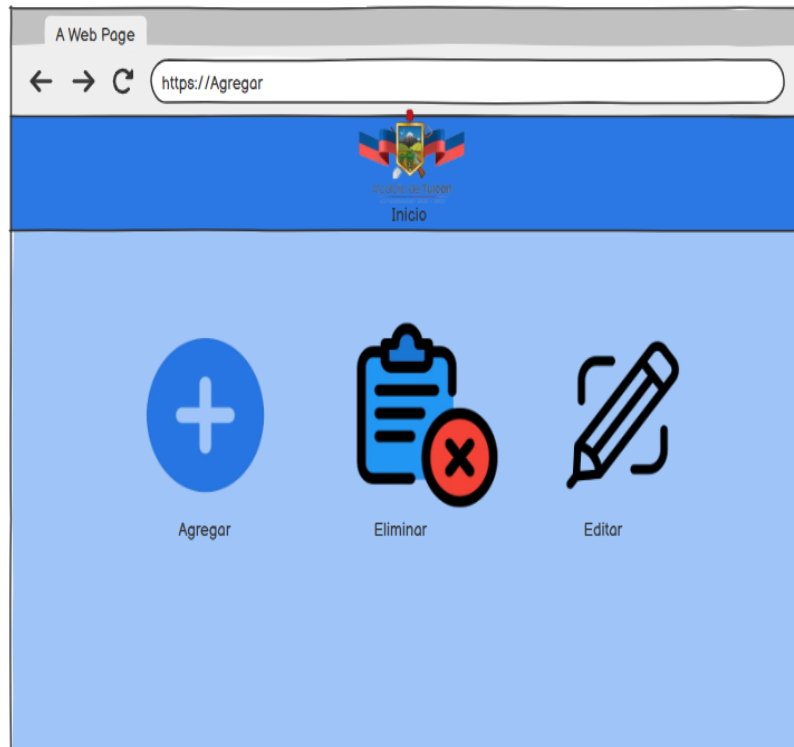


Figura 27. Prototipo pantalla agregar información

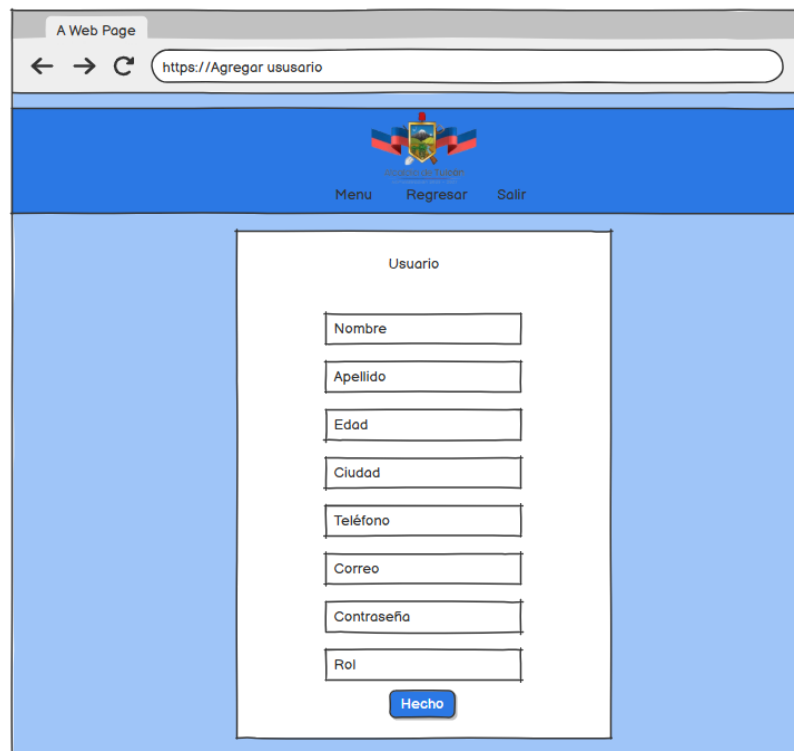


Figura 28. Prototipo pantalla agregar nuevo usuario

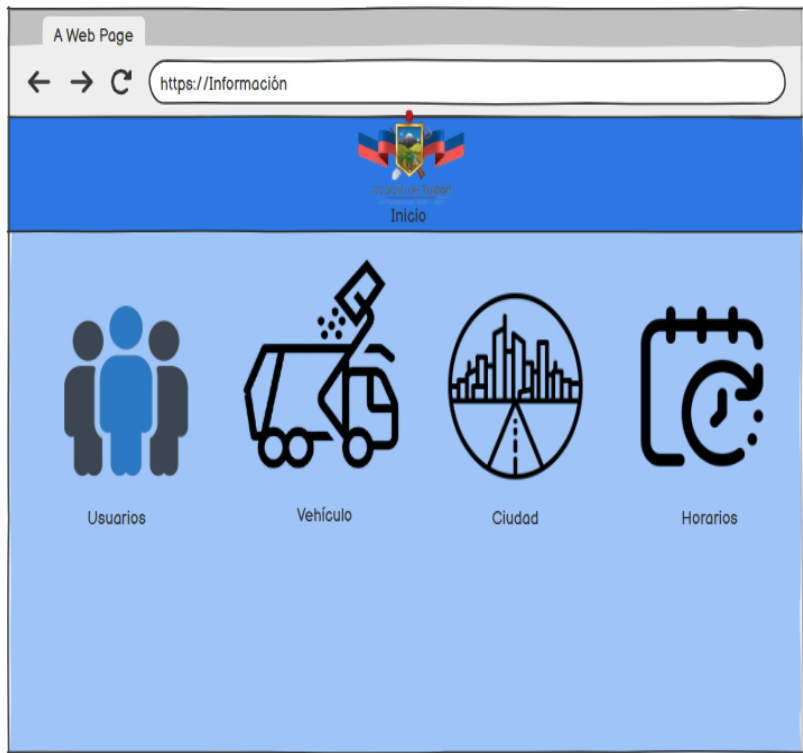


Figura 29. Prototipo pantalla ver información

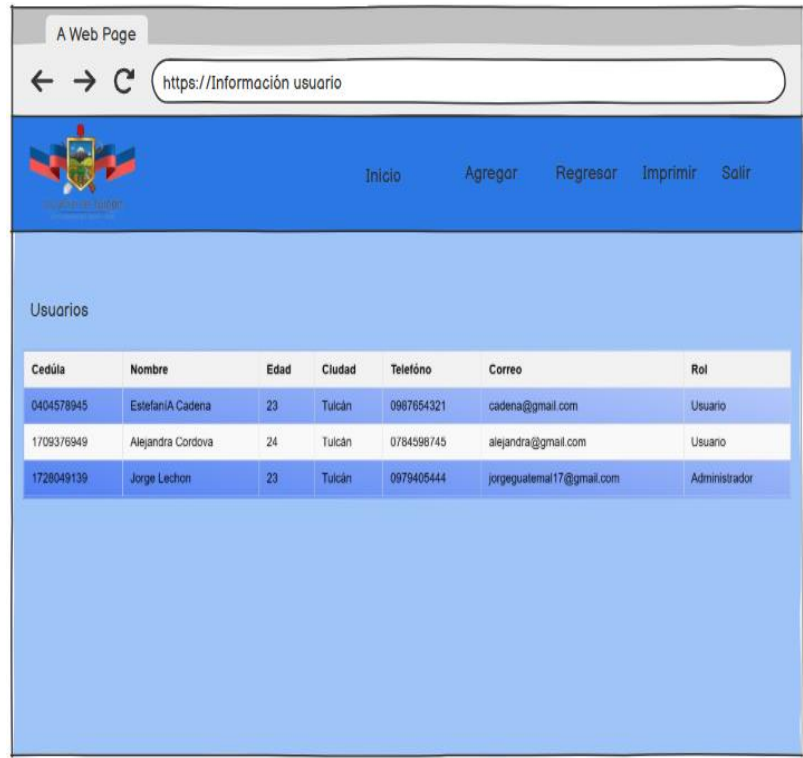


Figura 30. Prototipo pantalla ver información de usuario

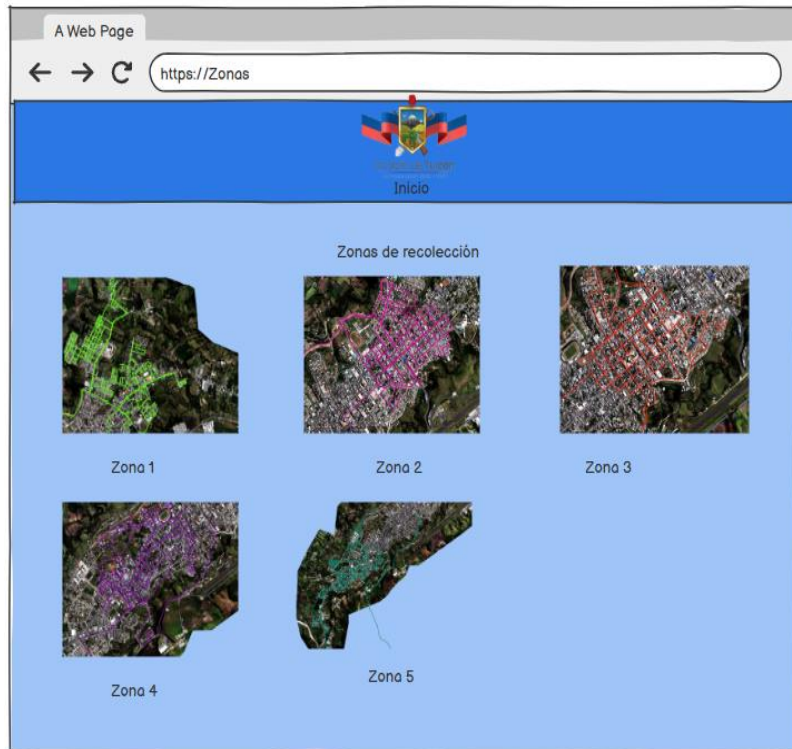


Figura 31. Prototipo pantalla zonas de recolección

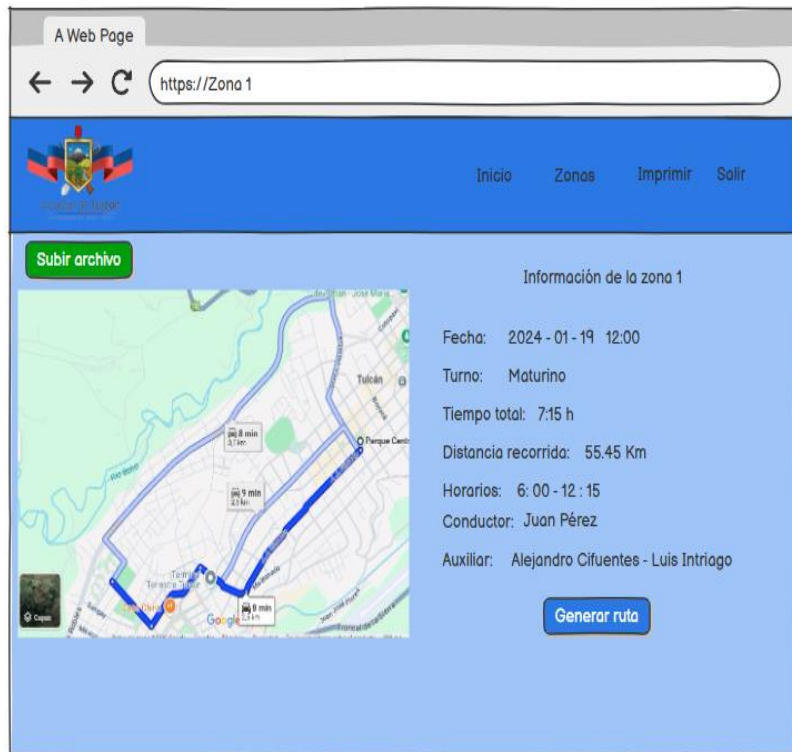


Figura 32. Prototipo pantalla generar ruta zona 1

4.1.10. Fase de Codificación

Server.js es el archivo que levanta el servidor, configura el comportamiento y organiza como responde a las peticiones.

```
const express = require('express');
const bodyParser = require('body-parser');
const { encriptarContraseña } = require('./src/helpers/authHelper');
const app = express();
const port = 3000;
const routes = require('./src/modules/index');
routes(app)

// Configuración de la conexión a la base de datos
const { connection, queryAsync } = require('./src/conexion')

// Middleware para parsear el cuerpo de las solicitudes
app.use(bodyParser.json());

// Servir archivos estáticos (tu frontend)
app.use(express.static('public'));
app.listen(port, () => {
  console.log(`Servidor corriendo en http://localhost:${port}`);
});
```

Figura 33. Código de levantamiento de servidor

- Conexión con bases de datos

```

const mysql = require("mysql2");
const connection = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: 'root',
  database: 'mydb_rutas'
});
connection.connect(err => {
  if (err) {
    console.error('Error conectando a la base de datos:', err);
    return;
  }
  console.log('Conectado a la base de datos');
});
const queryAsync = (consulta, params) => {
  return new Promise((resolve, reject) => {
    connection.query(consulta, params, (err, results) => {
      if (err) {
        reject(err);
      } else {
        resolve(results);
      }
    });
  });
};
module.exports = {
  connection,
  queryAsync
};

```

Figura 34. Código conexión base de datos MySQL

```

JS conexionNeo4j.js > [⌘] driver
const neo4j = require('neo4j-driver'); 554.1k (gzipped: 104.9k)

const user = 'neo4j';
const password = 'afAF^73DSF3^2024';
const host = 'neo4j://localhost:7687';

const driver = neo4j.driver(
  |   host,
  |   neo4j.auth.basic(user, password)
);

module.exports = driver

```

Figura 35. Código conexión base de datos Neo4j

```

const axios = require('axios') 61.1k (gzipped: 22.7k)

const osmRouteApi = axios.create({
  |   baseURL: 'https://router.project-osrm.org/route/v1/driving'
})

module.exports = {
  |   osmRouteApi
}

```

Figura 36. Código conexión API Open Street Maps

- CRUD

```

router.post('/registro', async (req, res) => {
  try {
    const { cedula, nombre, edad, ciudad, telefono, correo, contrasena, confirmarContrasena, fkRol } = req.body;
    if (!correo || !contrasena || !confirmarContrasena || !nombre || !cedula || !fkRol) {
      return res.status(400).json({ mensaje: 'Faltan campos obligatorios' });
    }
    if (contrasena !== confirmarContrasena) {
      return res.status(400).json({ mensaje: 'Las contraseñas no coinciden' });
    }

    //Buscar coincidencias en la cedula
    const consulta = 'SELECT * FROM usuarios WHERE Cedula = ?';
    const resultado = await queryAsync(consulta, [cedula]);
    if (resultado.length > 0) {
      return res.status(400).json({ mensaje: 'Ya existe un usuario con esta cedula' });
    }

    //Buscar coincidencias en el correo
    const consultaCorreo = 'SELECT * FROM usuarios WHERE correo = ?';
    const resultadoCorreo = await queryAsync(consultaCorreo, [correo]);
    if (resultadoCorreo.length > 0) {
      return res.status(400).json({ mensaje: 'Ya existe un usuario con este correo' });
    }

    // Encriptar contraseña
    const contrasenaEncriptada = await encriptarContrasena(contrasena);
  }
}

```

Figura 37. Código registro de usuario

```

router.get('/login', async (req, res) => {
  const { correo, contrasena } = req.query;

  if (!correo || !contrasena) {
    return res.status(400).json({ mensaje: 'Faltan campos obligatorios' });
  }

  const consultaCorreo = 'SELECT rol.rol, usu.contrasena, usu.correo FROM roles rol INNER JOIN usuarios usu ON rol.rol = usu.fkRol';
  const resultadoCorreo = await queryAsync(consultaCorreo, [correo]);
  if (resultadoCorreo.length === 0) {
    return res.status(403).json({ mensaje: 'Usuario o contraseña incorrectos' });
  }

  const usuario = resultadoCorreo[0];
  const contrasenaValida = await compararContrasena(contrasena, usuario.contrasena);
  if (!contrasenaValida) {
    return res.status(403).json({ mensaje: 'Usuario o contraseña incorrectos' });
  }

  // Generar token
  const data = generarToken({ correo: usuario.correo }, 3600);

  res.json({
    data: {
      ...data,
      rol: usuario.rol
    },
    mensaje: 'Usuario ingresó correctamente'
  });
}

```

Figura 38. Código autenticar usuario por contraseña y correo

```

// Usuario
app.post('/usuarios', async(req, res) => {
  const { campo1, campo2, campo3, campo4, campo5, campo6, campo7, rol } = req.body;

  //Buscar coincidencias en la cedula
  const consulta = 'SELECT * FROM usuarios WHERE Cedula = ?';
  const resultado = await queryAsync(consulta, [campo1]);
  if(resultado.length > 0){
    return res.status(400).json({ mensaje: 'Ya existe un usuario con esta cedula' });
  }
  //Buscar coincidencias en el correo
  const consultaCorreo = 'SELECT * FROM usuarios WHERE correo = ?';
  const resultadoCorreo = await queryAsync(consultaCorreo, [campo6]);
  if(resultadoCorreo.length > 0){
    return res.status(400).json({ mensaje: 'Ya existe un usuario con este correo' });
  }

  const contraseñaEncriptada = await encriptarContraseña(campo7);

  const query = 'INSERT INTO usuarios (Cedula, nombre, edad, ciudad, telefono, correo, contraseña, fk_rol) VALUES (';
  connection.query(query, [campo1, campo2, campo3, campo4, campo5, campo6, contraseñaEncriptada, rol], (err, result) => {
    if (err) {
      console.error('Error ejecutando la consulta:', err);
      res.status(500).send('Error en la consulta');
      return;
    }
  });
});

```

Figura 39. Código insertar datos MySQL

```

//USUARIO
app.get('/verusuarios', (req, res) => {
  const sql = 'SELECT usuarios.cedula, usuarios.nombre, usuarios.edad, usuarios.ciudad, usuarios.telefono, usuarios.rol FROM usuarios';
  connection.query(sql, (error, results) => {
    if (error) {
      return res.status(500).send('Error al obtener los datos');
    }
    res.json(results);
  });
});

```

Figura 40. Código consulta visualizar datos MySQL

```

//actualizar
// Ruta para actualizar un usuario por cédula
app.put('/editar-usuario/:cedula', async(req, res) => {
  const { cedula } = req.params;
  const { nombre, edad, ciudad, telefono, correo, contrasena, rol } = req.body;
  if (contrasena === '') {
    const query = 'UPDATE usuarios SET nombre = ?, edad = ?, ciudad = ?, telefono = ?, correo = ?, fk_rol = ? WHERE cedula = ?';
    connection.query(query, [nombre, edad, ciudad, telefono, correo, rol, cedula], (err, result) => {
      if (err) throw err;

      if (result.affectedRows > 0) {
        res.send({ message: 'Usuario actualizado con éxito' });
      } else {
        res.status(404).send({ message: 'Usuario no encontrado' });
      }
    });
  } else {
    const contrasenaEncriptada = await encriptarContrasena(contrasena);

    const query = 'UPDATE usuarios SET nombre = ?, edad = ?, ciudad = ?, telefono = ?, correo = ?, contrasena = ? WHERE cedula = ?';
    connection.query(query, [nombre, edad, ciudad, telefono, correo, contrasenaEncriptada, rol, cedula], (err, result) => {
      if (err) throw err;

      if (result.affectedRows > 0) {
        res.send({ message: 'Usuario actualizado con éxito' });
      } else {
        res.status(404).send({ message: 'Usuario no encontrado' });
      }
    });
  }
});

```

Figura 41. Código editar datos MySQL

```

// Ejecutar la consulta DELETE
const sql = `DELETE FROM Usuarios WHERE Cedula = ?`;
connection.query(sql, [Cedula], (err, result) => {
  if (err) {
    console.error('Error al ejecutar la consulta:', err);
    return res.status(500).json({ message: 'Ocurrió un error al eliminar Usuario' });
  }

  if (result.affectedRows > 0) {
    res.json({ message: `Usuario con Cedula ${Cedula} eliminado correctamente.` });
  } else {
    res.json({ message: `No se encontró un Usuario con esta cedula ${Cedula}.` });
  }
});
});

```

Figura 42. Código eliminar datos MySQL

- Roles

```
> JS validarTokenPrivado.js
validarTokenPrivado();
validarAdmin();

async function validarTokenPrivado() {
  const token = localStorage.getItem('token');
  if (!token) {
    localStorage.removeItem('token');
    localStorage.removeItem('rol');
    window.location.href = '/index.html';
  }
  try {
    const response = await fetch(`${pathUrl}/auth/validar?token=${token}`, {
      method: 'GET'
    });
    const { status } = await response;
    const { mensaje } = await response.json();

    if(status !== 200) {
      alert(mensaje);
      localStorage.removeItem('token');
      localStorage.removeItem('rol');
      window.location.href = '/index.html';
    }
  } catch (error) {
    console.error('Error validando el token:', error);
  }
}

function validarAdmin() {
```

Figura 43. Código autenticación rol administrador

```

JS validarTokenPublico.js
validarTokenPublico();

async function validarTokenPublico() {
  const token = localStorage.getItem('token');
  try {
    const response = await fetch(`${pathUrl}/auth/validar?token=${token}`, {
      method: 'GET'
    });
    const { status } = await response;

    if(status === 200) {
      window.location.href = '/menu.html';
    }
  } catch (error) {
    console.error('Error validando el token:', error);
  }
}

```

Figura 44. Código autenticación rol usuario

- Creación de relaciones en la API Open Street Maps

```

const express = require('express');
const {
  getPointsByZone: getPointsByZoneNeo4j,
  setPoint: setPointNeo4j,
  setRelationPoint: setRelationPointNeo4j,
  getRelationPointsByZone
} = require( './models/neo4jPuntos');

const { getPuntosByZona, getPuntosByZonaNoIds, getPuntosOrdenados } = require('./models/puntos');
const { setRutas, getRutasByViaje } = require('./models/rutas');
const { setViaje, getViajesByZona, getViajesById } = require('./models/viajes');
const { setViajeRutasRel } = require('./models/viajeRutasRel');
const { getUsuariosByRol } = require('./models/usuarios');
const { getVehiculos } = require('./models/vehiculo');

const { osmRouteApi } = require('./api/osmApi');
const { addSecondsToTime, showTime, showDistance, getTimeZone, getDateFormat } = require('./helpers/dateHelper');

const router = express.Router();
router.use(express.json());

```

Figura 45. Código creación relaciones entre puntos de las zonas

```

// Creacion de rutas
router.post('/create', async(req, res) => {
  const {zone, notPoints, startTime, timeService, driver, truck, auxs, turn} = req.body
  const startPoint = await getPuntosOrdenados(zone)
  let startPointObj = startPoint[0]

  const pointsNotFound = [startPointObj.id, ... notPoints]

  const points = await getPuntosByZonaNoIds(String(zone), pointsNotFound) // Obtener las ubicaciones de una zona
  const relationsPoints = await getRelationPointsByZone(String(zone), 'OSM') // Obtener las relaciones de una zona

  const allPoints = JSON.parse(JSON.stringify(points))
  // Creacion de la ruta
  const routes = route(points.filter(point => point.status === 1), allPoints, relationsPoints, [{point: startPointObj}])

  const routesInsert = []
  let consecutive = 1
  for(const {properties, point} of routes){
    if(!point) continue

    const {distance, time, codedPolylines} = properties

    let startTimeRoute = ''
    let endTimeRoute = ''
    const service = timeService * 60
    if(consecutive === 1) {
      startTimeRoute = startTime
    }else{

```

Figura 46. Código creación de rutas

```

// Funcion recursiva de la ruta
const route = (points, allPoints, relations, routes) => {
  if(points.length === 0) {
    return routes
  }
  //Buscar el punto mas cercano a partir de la ultima ubicacion
  const ultPoint = routes[routes.length - 1].point
  const pointStarts = relations.filter((point) => point.start.id === ultPoint.id)
  //Buscar la relacion mas corta
  const nextPointNeo4j = getShortestRoute(pointStarts, points, allPoints, ultPoint)
  if(nextPointNeo4j) {
    //Agregar el punto a la ruta
    const nextPointIndex = points.findIndex(point => point.id === nextPointNeo4j.end.id)
    routes.push({
      point: points[nextPointIndex],
      properties: nextPointNeo4j.properties
    })
    //Eliminar el punto de la lista de puntos
    removeObjectToArr(points, points[nextPointIndex])
  }else{
    console.log('No se encontro una ruta')
    removeObjectToArr(points, points[0])
  }
}

```

Figura 47. Código función recursiva

```

const getShortestRoute = (relations, points, allPoints, point) => {
  //Manejar la ruta por distancia
  let distanceObj = undefined
  distanceObj = relations.find((relation) => relation.end.id === points[0].id)
  if(!distanceObj) return undefined

  // Hace búsqueda de toda la información
  for(const relation of relations) {
    const isExist = points.some(pointF => pointF.id === relation.end.id)
    if(!isExist) continue

    const pointRelation = allPoints.find(pointF => pointF.id === relation.end.id)
    if (!pointRelation) continue;

    // Determinar si es de mayor prioridad
    let isPriority = (points[0].prioridad || 0) >= (pointRelation.prioridad || 0);

    // Asegurarse de que la distancia esté bien definida y sea un número
    if (!relation.properties || isNaN(Number(relation.properties.distance))) continue;

    // Convertir la distancia a número antes de compararla
    const relationDistance = Number(relation.properties.distance);

    // Comparar la distancia si la prioridad lo permite
    if (isPriority && relationDistance < distanceObj.properties.distance) {
      distanceObj = relation; // Actualizar si la relación actual es mejor
    }
  }
}

```

Figura 48. Código manejo de prioridades en los puntos

4.1.11. Presentación del aplicativo

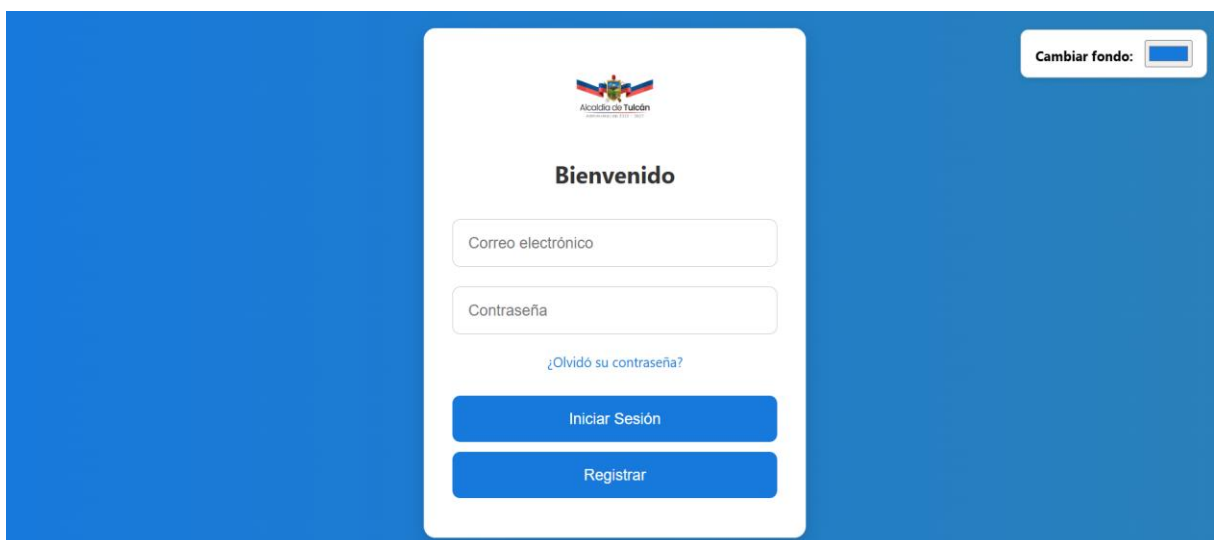


Figura 49. Interfaz inicial

Alcaldía de Tulcán

Inicio

Registrar Usuario

Cédula

Nombre

Apellido

Edad

Tulcán

Teléfono

Figura 50. Interfaz registro usuario

Recuperación de Contraseña

Código generado:

Válido por 15 minutos

Correo registrado

Generar Código

Código recibido

Nueva contraseña

Cambiar Contraseña

← Volver al Login

Figura 51. Interfaz recuperación de contraseña

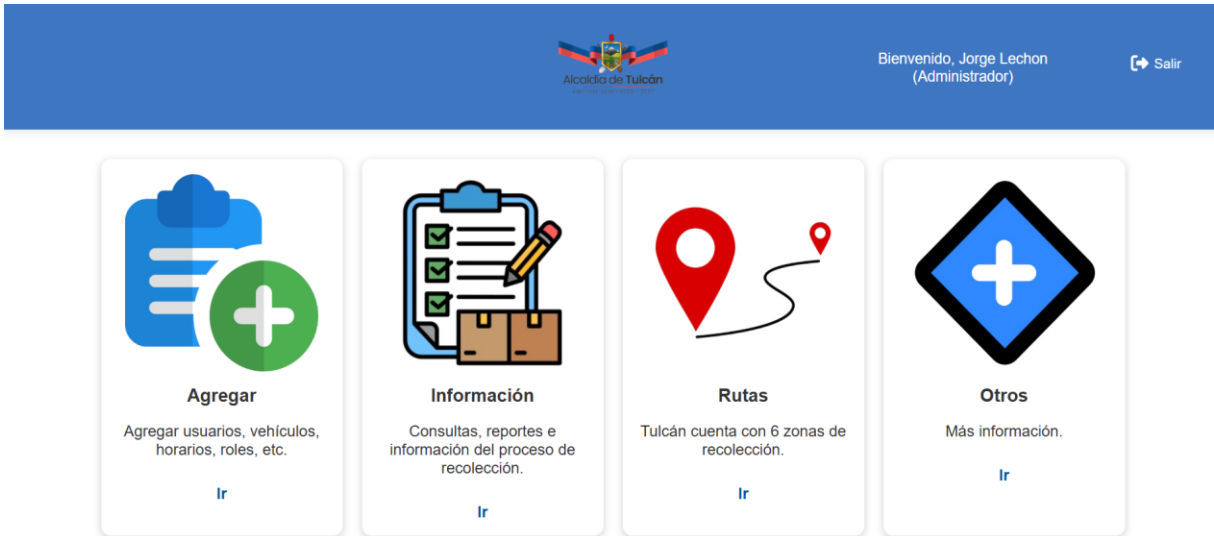


Figura 52. Interfaz módulos

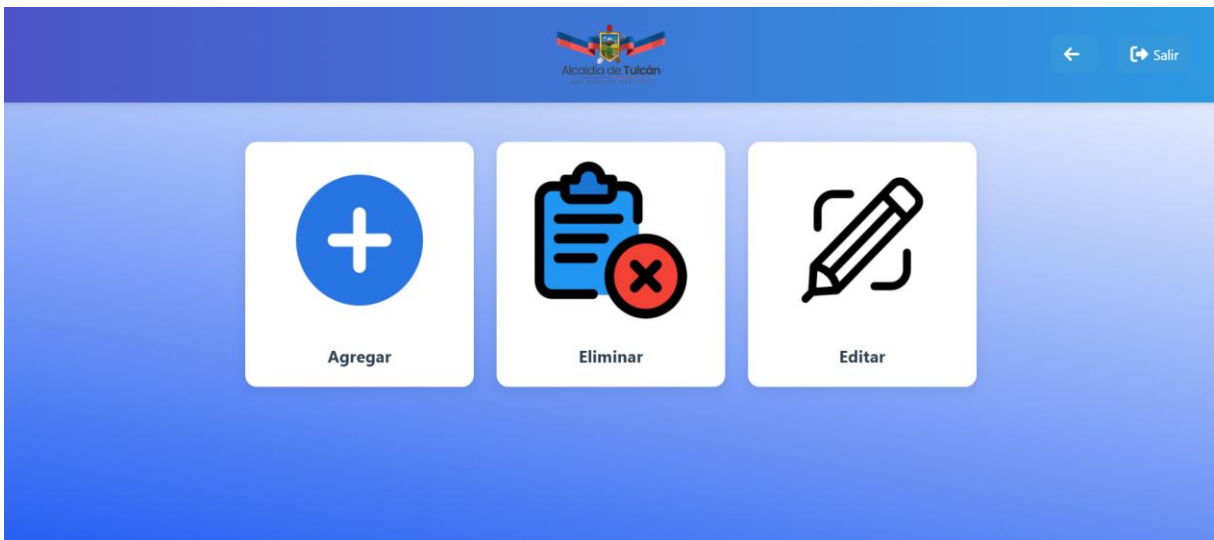


Figura 53. Interfaz módulo Agregar

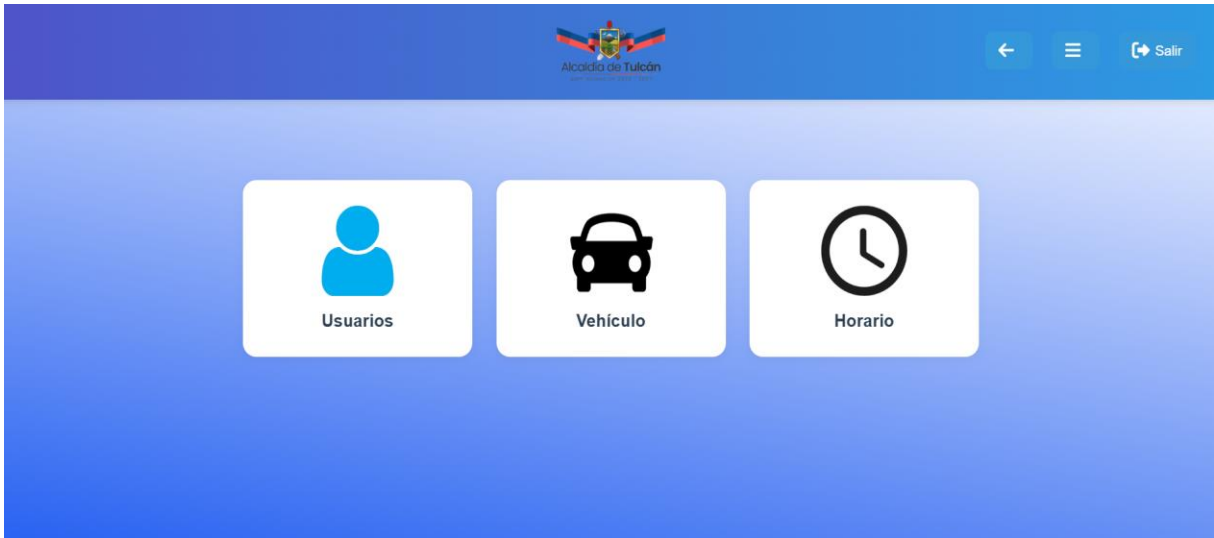


Figura 54. Interfaz módulo información

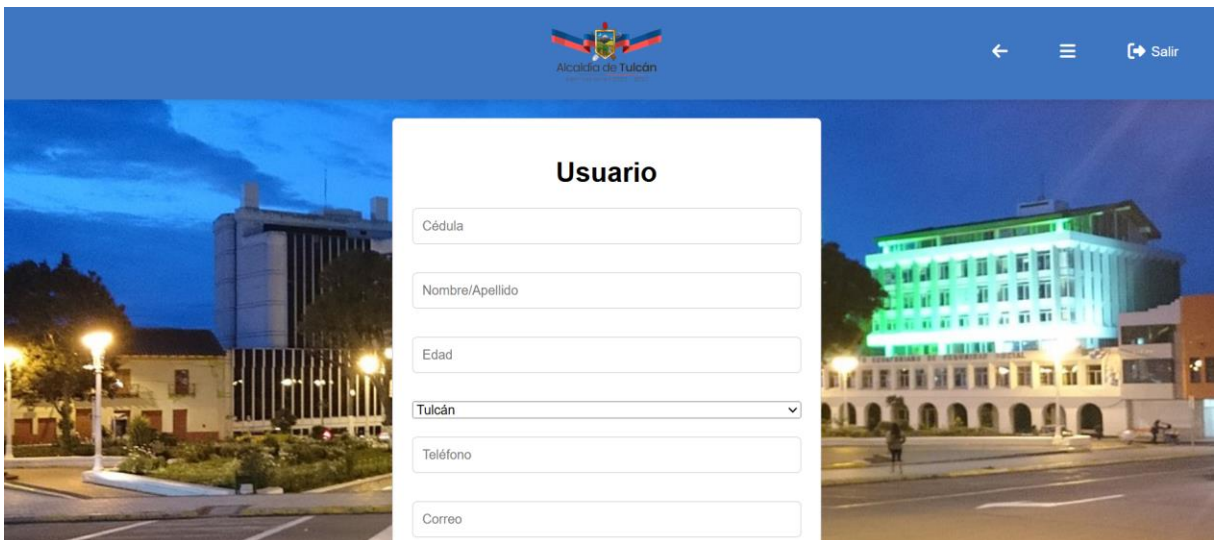


Figura 55. Interfaz registro usuario

Cedula	Nombre	Edad	Ciudad	Teléfono	Correo	Rol
0404578945	Estefanía Cadena	23	Tulcán	0987654321	cadena@gmail.com	Usuario
1709376949	Alejandra Cordova	24	Tulcán	0784598745	alejandra@gmail.com	Usuario
1728049139	Jorge Lechon	23	Tulcán	0979405444	jorgeguatemal17@gmail.com	Administrador

Figura 56. Interfaz visualizar usuarios

Figura 57. Interfaz módulo rutas

Información de la ruta 2

Selector de ruta:
2025-01-19 - Matutino

Fecha de la ruta: 2025-01-19 16:36:14

Turno: Matutino

Tiempo total: 05.07 hrs.

Distancia recorrida: 44.033 km.

Horario: 06:00 - 10:09 hrs.

Conductor: Alejandro Cifuentes

Auxiliares: Wilmer Colimba, Alejandra Cordova

Camión: CMA-1028

Figura 58. Interfaz módulo rutas

4.1.12. Fase de Pruebas

4.2. DISCUSIÓN

La presente investigación propone un sistema de gestión de rutas para la recolección de desechos sólidos, apoyado en el algoritmo de Dijkstra, con el objetivo de optimizar los trayectos y mejorar la eficiencia operativa. En comparación con el estudio de Castro (2023) , que implementó el mismo algoritmo en un sistema de transporte público urbano, ambos proyectos comparten una base metodológica sólida al emplear grafos dirigidos representados mediante matrices de adyacencia. Sin embargo, mientras Castro se enfocó en la movilidad de pasajeros, mientras esta investigación traslada el enfoque hacia un servicio esencial como la gestión de residuos, adaptando el algoritmo a la realidad operativa del municipio de Tulcán.

Por otro lado, Sánchez (2021) destaca la relevancia del uso de sistemas de información geográfica (SIG), como ArcGIS, para mejorar la planificación y visualización de rutas. En su estudio, los SIG permitieron una representación geoespacial detallada de las zonas de recolección, facilitando la toma de decisiones basada en mapas interactivos que muestran puntos críticos, accesos y obstáculos del entorno urbano. En contraste, el sistema desarrollado en esta investigación se centró en la optimización matemática de las rutas a través del algoritmo de Dijkstra, sin incorporar herramientas SIG. No obstante, los beneficios documentados por Sánchez evidencian una oportunidad valiosa para futuras mejoras del sistema, ya que la integración de SIG permitiría complementar el cálculo de rutas óptimas con una visualización georreferenciada más precisa y dinámica. De este modo, el sistema propuesto podría evolucionar hacia una herramienta más robusta, capaz de ofrecer tanto eficiencia en los recorridos como una interfaz visual amigable para los operadores y autoridades responsables de la gestión de desechos sólidos.

Por otro lado, Talha, Rizwan y Rizwan (2024) presentaron un estudio centrado en la identificación de rutas óptimas para la instalación de oleoductos y gasoductos en Pakistán, utilizando Sistemas de Información Geográfica (SIG) como herramienta clave. Aunque su contexto difiere del de esta investigación, ambos comparten el propósito de optimizar trayectos mediante el análisis técnico del terreno. Mientras el sistema propuesto en este trabajo se enfoca en mejorar la recolección de residuos

sólidos en un entorno urbano como Tulcán, el estudio de Talha et al. resalta cómo la integración de tecnologías geoespaciales puede fortalecer la planificación y reducir riesgos, ofreciendo una referencia valiosa para futuras mejoras del sistema de gestión de rutas.

Adicionalmente, Márquez y López (2020) exploran algoritmos más complejos como colonia de hormigas y genéticos, los cuales ofrecen ventajas en escenarios dinámicos o de mayor escala. No obstante, el presente estudio optó por Dijkstra debido a su bajo costo computacional y eficiencia en entornos urbanos medianos con pesos positivos, como el de Tulcán, lo cual permite un equilibrio adecuado entre rendimiento y simplicidad.

Por su parte, la investigación de Aragón (2021), aunque centrada en el diseño de rutas accesibles para personas con discapacidad visual, introduce un enfoque multicriterio que abre nuevas posibilidades para el desarrollo de sistemas de planificación. Este enfoque sugiere la incorporación de variables adicionales, como la seguridad vial, el estado de la infraestructura o los niveles de tráfico, con el fin de generar rutas no solo más cortas, sino también más seguras y funcionales. En comparación, el sistema propuesto en esta investigación actualmente se basa en el criterio de distancia mínima utilizando el algoritmo de Dijkstra. No obstante, se reconoce la importancia de considerar estos factores complementarios, por lo que su integración se plantea como una mejora futura para enriquecer la calidad y eficiencia de la planificación de rutas.

En conjunto, los antecedentes analizados refuerzan la solidez técnica y funcional del sistema propuesto, destacando que la aplicación del algoritmo de Dijkstra, combinado con el modelado de grafos y herramientas como OpenStreetMaps o Google Maps, permite una solución efectiva y adaptable. A diferencia de estudios centrados en funciones específicas o en otras industrias, este proyecto responde a una necesidad municipal concreta, ofreciendo una herramienta que es de gran aporte a la recolección de residuos sólidos con beneficios en sostenibilidad y planificación operativa.

V. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- La fundamentación bibliográfica fue clave para identificar y sustentar las variables técnicas necesarias para el desarrollo del sistema de rutas óptimas. Los antecedentes revisados permitieron reconocer la aplicabilidad de algoritmos como Dijkstra, así como el uso de grafos y tecnologías de programación adecuadas, lo cual fortaleció el diseño del sistema propuesto.
- La investigación del proceso de recolección permitió identificar datos claves como frecuencias, horarios, cobertura de rutas y dificultades operativas. Esta información, recolectada tanto de forma cualitativa como cuantitativa, fue fundamental para modelar correctamente el sistema y justificar su necesidad en función de las condiciones reales del municipio.
- Tras comparar diferentes algoritmos, como Bellman-Ford, Floyd-Warshall y Dijkstra, se determinó que este último es el más adecuado para el contexto urbano de Tulcán, debido a su menor complejidad computacional, precisión en grafos con pesos positivos y eficiencia comprobada en la búsqueda de rutas óptimas.
- El sistema informático desarrollado permite calcular rutas de menor costo utilizando el algoritmo de Dijkstra, lo que mejora la eficiencia operativa del proceso de recolección. Además, al estar construido con tecnologías como JavaScript y estructuras modulares, es escalable, adaptable y capaz de integrarse con futuras herramientas de georreferenciación, lo que garantiza su sostenibilidad a largo plazo.

5.2. RECOMENDACIONES

- Se recomienda continuar con la revisión y actualización bibliográfica sobre rutas óptimas y tecnologías emergentes aplicadas a la recolección de desechos, a fin de enriquecer futuras mejoras del sistema con enfoques más innovadores y adaptados a nuevas necesidades.

- Para reforzar la calidad del sistema, es aconsejable realizar estudios periódicos del proceso operativo de recolección de desechos, manteniendo una base de datos actualizada con información cualitativa y cuantitativa que permita ajustar rutas de manera dinámica.
- Se sugiere explorar la integración de algoritmos multicriterio en versiones futuras, incorporando variables como el estado de la vía, el nivel de tráfico o zonas de riesgo, lo cual permitiría una planificación más realista y segura, superando la lógica del camino más corto.
- Es fundamental mantener una arquitectura de software abierta y modular, que facilite la incorporación de tecnologías complementarias como sistemas de información geográfica (SIG) o sensores IoT para monitoreo en tiempo real, lo cual potenciaría la precisión geoespacial del sistema.
- Se recomienda mantener una comunicación continua con las entidades municipales involucradas, alineando expectativas, requerimientos y funcionalidades de la herramienta, con el fin de asegurar que el sistema se ajuste adecuadamente a la operativa del servicio público.

VI. REFERENCIAS BIBLIOGRÁFICAS

- Adhikary, A., Kar, A., & Gupta, A. (2021). Multi-criteria decision-making model for solid waste collection route optimization. *Resources, Conservation and Recycling. Resconect*, 146, 34-47. <https://doi.org/10.1016>
- Aguayo Moreno, E. (2008). *Desarrollo de un sistema de localización de rutas óptimas entre dos puntos geográficos [Tesis de pregrado]*. Universidad San Francisco de Quito. <https://doi.org/https://repositorio.usfq.edu.ec/handle/23000/548>
- Aguayo, E., & Moreno, M. (Mayo de 2008). *Repositorio Digital USFQ*. Retrieved 23 de 04 de 2025, from *Desarrollo de un sistema de localización de rutas óptimas entre dos puntos geográficos*: <https://192.188.53.14/handle/23000/548>
- Ahamed, H. (2025, Abril 13). *Simplilearn*. Retrieved Mayo 2, 2025, from *Best Programming Languages in 2025 - Which One to Learn?:* <https://www.simplilearn.com/best-programming-languages-start-learning-today-article>
- Alaimo Labs. (2019). *Roles y Responsabilidades del Equipo Scrum*. Alaimo Labs: <https://alaimolabs.com/es/self-learning/scrum/roles-del-equipo-scrum-desarrolladores-scrum-master-y-product-owner?srltid=AfmBOor2UMIAOHEECxst3860prAvisjQse3x1DtbJ24xddqMMm0iAE4R>
- All, M. (2024 de Julio de 2024). Retrieved 24 de Abril de 2025, from *¿Qué es PostgreSQL?:* <https://www.datacamp.com/es/blog/what-is-postgresql-introduction>
- Altenwald. (2024). *Grafos (I): Fundamentos Básicos*. Altenwald Blog: <https://altenwald.org/2011/10/02/grafos-fundamentos-basicos/>
- Amazon Web Services. (2022). *¿En qué consiste Scrum?-Explicación sobre la metodología Scrum*. Amazon Web Services: <https://aws.amazon.com/es/what-is/scrum/>
- Amazon Web Services. (2022). *¿Qué es JavaScript (JS)?* Amazon Web Services: <https://aws.amazon.com/es/what-is/javascript/#:~:text=JavaScript%20es%20un%20lenguaje%20de,usuario%20de%20un%20sitio%20web.>
- Amazon Web Services. (2022). *¿Qué es JavaScript (JS)?* Amazon Web Services: <https://aws.amazon.com/es/what-is/javascript/>

- Amazon Web Services. (2024). *Amazon Web Services*. Retrieved Abril 24, 2025, from Amazon Neptune: <https://aws.amazon.com/es/neptune/>
- Ambientum. (28 de noviembre de 2022). *Gestión de residuos sólidos urbanos - Ambientum*. Retrieved 24 de Abril de 2025, from Ambientum Portal Lider Medioambiente: https://www.ambientum.com/enciclopedia_medioambiental/suelos/consideracion_general_a_la_gestion.asp
- Angles, R., & Guitierrez, C. (2018). *Foundations of modern graph database systems*. ACM Computing Surveys. <https://doi.org/10.1145/3231596>
- Anil, U. (2024, Febrero 9). *Medium*. Retrieved Abril 24, 2025, from How Floyd-Warshall Algorithm Works: <https://medium.com/@umitanilkilic/floyd-warshall-algorithm-c65f274352a7>
- Anton, J., Bravo, J., Arcila, J., & Tuesta, V. (2021). Planificador de rutas para el recojo de desechos sólidos utilizando el algoritmo de Dijkstra. *Revista científica del señor de Sipán*, 8(2). <https://doi.org/10.26495/icti.v8i2.1918>
- APD. (2024, abril 09). *Cómo aplicar la metodología Scrum y qué es el método Scrum*. APD España: <https://www.apd.es/metodologia-scrum-que-es/>
- APD. (2024, abril 09). *Cómo aplicar la metodología Scrum y qué es el método Scrum*. <https://www.apd.es/metodologia-scrum-que-es/>
- Aragón, M. (28 de septiembre de 2021). *Planificación de rutas accesibles para personas con discapacidad visual utilizando programación lineal entera y el algoritmo de Dijkstra*. <https://doi.org/20.500>
- Arango, A. (10 de 01 de 2024). *Logístiko*. Retrieved 23 de 04 de 2025, from Gestión de rutas de transporte: <https://logistiko.es/blog/gestion-de-rutas-de-transporte/>
- Arboleda, A. (2020, marzo 13). *Conoce las funciones de un Product Owner y su importancia en los proyectos ágiles*. Rockcontent: <https://rockcontent.com/es/blog/product-owner/>
- Arsys. (2021, julio 23). *Mejores extensiones de VSCode para JavaScript, HTML, PHP y Python*. Arsys: <https://www.arsys.es/blog/extensiones-vscode>
- arsys.es. (2019, abril 5). *Axios Javascript: analizamos las características de este ligero cliente HTTP*. arsys.es: <https://www.arsys.es/blog/axios>
- Asensio, A. (2017, Septiembre 14). *Paradigma*. Retrieved Abril 24, 2025, from ¿Conoces ArangoDB?: <https://www.paradigmadigital.com/dev/hola-conoces-arangodb/>
- Atlassian. (2016). *Qué es scrum y cómo empezar*. Atlassian. <https://doi.org/https://www.atlassian.com/es/agile/scrum>

- Attenborough, M. (2003). *ScienceDirect*. Retrieved Abril 24, 2025, from Teoría de Grafos: https://www-sciencedirect-com.translate.goog/topics/computer-science/weighted-graph?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc
- Axios Docs. (2025). ¿Qué es Axios? Axios Docs: <https://axios-http.com/es/docs/intro>
- Bagus, D. (2020, Julio 27). *Medium*. Retrieved Abril 24, 2025, from How Dijkstra's Algorithm Works?: <https://medium.com/@darubagus/how-dijkstras-algorithm-works-8b249c2431a1>
- Barragán, A. (2023, septiembre 14). *Vue Axios: Cómo consumir y manipular datos de una API REST*. OpenWebinars: <https://openwebinars.net/blog/vue-axios-como-consumir-y-manipular-datos-de-una-api-rest/#:~:text=Qu%C3%A9%20es%20Axios%20y%20qu%C3%A9%20ventajas%20ofrece%20en%20el%20proceso,-Axios%20es%20una&text=Las%20principales%20ventajas%20de%20esta,y%20entornos%20ba>
- Bejtullahu, D. (2022). *PECB*. Retrieved 24 de Abri de 2025, from ISO/IEC 25000 - Requisitos y evaluación de la calidad del software: <https://pecb.com/article/isoiec-25000---software-quality-requirements-and-evaluation---square>
- Betania, V. (2023, mayo 03). *Qué es un SGBD: Guía completa sobre los sistemas de gestión de bases de datos*. Retrieved Abril 29, 2024, from Hostinger Tutoriales: <https://www.hostinger.es/tutoriales/sghbd>
- Bhaskar, R. (2016, Abril 8). *Medium*. Retrieved Abril 24, 2025, from How and when to use OrientDB: A study by HelloClass: <https://medium.com/@helloclassmedia/how-and-when-to-use-orientdb-a-study-by-helloclass-b84404e2bc9>
- bignumber.js. (2025). *bignumber.js*. [bignumber.js: https://mikemcl.github.io/bignumber.js/](https://mikemcl.github.io/bignumber.js/)
- Bock, T. (2025, Febrero 11). *Displayr*. Retrieved Abril 24, 2025, from ¿Qué es una matriz de distancias?: <https://www.displayr.com/what-is-a-distance-matrix/>
- Bombase, R. (2017, Agosto 30). *Medium*. Retrieved Abril 24, 2025, from ISO 9126 Software Quality Characteristics: <http://medium.com/@ryanbombase05/output-of-the-week-ceba563b0799>
- Bravo, J., Anton, J., Arcila, J., & Tuesta, V. (01 de Octubre de 2021). *Universidad Señor Sipán*. <https://doi.org/10.26495>
- Brodowicz, M. (2024, junio 23). *The Impact of Pollution on Urban Environments and Public Health*. Aithor: <https://aithor.com/essay-examples/the-impact-of-pollution-on-urban-environments-and-public-health>
- Brudvig, G., & Aspelin, A. (2019). Can citizen science improve the monitoring of solid waste management? A case study from rural India. *Vasman*, 89, 3-12. <https://doi.org/10.1016>

- Buhrkal, K., Larsen, A., & Ropke, S. (2012). The waste collection vehicle routing problem with time windows in a city logistics context . *Procedia*, 39, 241-254. <https://doi.org/10.1016>
- Calvo, C. (5 de Julio de 2024). *Euroinnova*. Retrieved 24 de Abril de 2025, from ¿Qué es la ingeniería en software?: <https://www.euroinnova.com/blog/que-es-la-ingenieria-en-software>
- Campani, D., Espinoza, P., & Sarafian, D. (2018). *Gestión integral de residuos sólidos urbanos*. Retrieved Abril 24, 2025, from <https://aidisnet.org/wp-content/uploads/2019/08/GESTION-INTEGRAL-DE-RESIDUOS-SOLIDOS-URBANOS-LIBRO-AIDIS.pdf>
- Cano, A., & Paéz, J. (2018). *Sistema de Transporte y logística*. Universidad Nacional.
- Castro Tinoco, O., & Hasimoto Beltrna, R. (2023). *Sistema de ruta óptima y plan de viaje para el sistema de transporte público*. Repositorio Institucional del Tecnológico Nacional de México (RI - TecNM). <https://doi.org/https://rinacional.tecnm.mx/jspui/handle/TecNM/6223>
- Castro, O. (21 de Abril de 2023). *Instituto Tecnológico Superior del Sur de Guanajuato*. Retrieved 24 de Abril de 2025, from Sistema de ruta óptima y plan de viaje para el sistema de transporte público: <https://rinacional.tecnm.mx/bitstream/TecNM/6223/1/TESIS%20S16120248%20CASTRO%20TINOCO%20OSWALDO.pdf>
- Catalina, D. (2022). *Qué es Node.js: Casos de uso comunes y cómo instalarlo*. Hostinger Tutoriales. <https://doi.org/https://www.hostinger.com/es/tutoriales/que-es-node-js>
- CEEI. (2022, diciembre 6). *JavaScript: características, beneficios y casos de uso*. CEEI: <https://ceeivalencia.emprenemjunts.es/?op=8&n=28660>
- Cepal.org. (2025). *Gobiernos Autónomos Descentralizados de Ecuador*. Cepal.org: <https://observatorioplanificacion.cepal.org/es/instituciones/gobiernos-autonomos-descentralizados-de-ecuador>
- Chinmay, Y. (23 de febrero de 2024). *Linkelind*. Retrieved 24 de Abril de 2025, from The Impact of Inefficient Waste Management: Exploring Environmental Consequences: <https://www.linkedin.com/pulse/impact-inefficient-waste-management-exploring-chinmay-yajnik-khmff>
- Córdova, J., & Rivas, M. (2020). *Gestión integral de residuos sólidos urbanos*. Trillas. <https://doi.org/9786071716949>
- Coronado, E. (2025). *Scribd*. Retrieved 24 de Abril de 2025, from Metodología Iconix: <https://es.scribd.com/presentation/218003718/Metodologia-Iconix-pptx>
- Crespo, A. (18 de Abril de 2018). *Excentia*. Retrieved 24 de Abril de 2025, from ISO 25000: La calidad del producto software: <https://www.excentia.es/iso-25000>

- Cuervo, W., & Ortiz, M. (2018, Septiembre 14). *Repositorio UPTC*. Retrieved Abril 25, 2025, from Caracterización de metodologías ágiles para el desarrollo de aplicaciones educativas móviles: <http://repositorio.uptc.edu.co/handle/001/5892>
- Dadedzi, K., & Wagner, S. (2014, mayo). *The Distance Matrix of a Graph*. Retrieved Abril 24, 2025, from ResearchGate: https://www.researchgate.net/publication/263622014_The_Distance_Matrix_of_a_Graph
- Darias, S. (2021, Octubre 18). *Intelequia*. Retrieved Abril 24, 2025, from ¿Qué es Microsoft SQL Server y para qué sirve?: <https://intelequia.com/es/blog/post/qu%C3%A9-es-microsoft-sql-server-y-para-qu%C3%A9-sirve>
- Dasgupta, S., & Papadimitriou, C. (2017). *Algorithms (2nd ed.)*. McGraw-Hill Education.
- Dassi, O. (2024, Enero 8). *Baeldung*. Retrieved Abril 24, 2025, from Introduction to the OrientDB Java APIs: <https://www.baeldung.com/java-orientdb>
- De Dios, M. A. (2024, septiembre 4). *Qué es Scrum y cómo funciona esta metodología de trabajo*. We are Marketing.com: <https://www.wearemarketing.com/es/blog/scrum-que-es-como-funciona.html>
- Del Pozo, H. (2019, Diciembre 31). *Consejo de Participación Ciudadana y Control Social*. Retrieved Abril 28, 2025, from Código Orgánico de Organización Territorial COOTAD: <chrome-extension://efaidnbnmnibpcajpcgclclefindmkaj/https://www.cpmccs.gob.ec/wp-content/uploads/2020/01/cootad.pdf>
- desarrolloweb.com. (2015). *Javascript*. desarrolloweb.com: <https://desarrolloweb.com/home/javascript>
- Deshpande, C. (2025, Abril 13). *Simplilearn*. Retrieved Abril 24, 2025, from The Best Guide to Know What Is React: <https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs>
- Docs MDN Web. (2025, febrero 7). *CSS grid layout*. https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_grid_layout
- DOCS, M. W. (2024). *Diseño de caja flexible CSS*. Mmdn web docs. https://doi.org/https://developer.mozilla.org/es/docs/Web/CSS/CSS_flexible_box_layout
- Donetonic. (2021, mayo 26). *Los 5 eventos en Scrum claves para el desarrollo de DoneTonic*: <https://donetonic.com/es/que-son-los-sprints-en-scrum/>
- Dongee. (2024, abril 7). *MySQL Workbench: Características, Ventajas y Desventajas*. Dongee: <https://www.dongee.com/tutoriales/mysql-workbench-caracteristicas-ventajas-y-desventajas/>

- Doshi, D., Jain, L., & Gala, K. (Abril de 2021). *ResearchGate*. Retrieved 24 de Abril de 2025, from Review of the Spiral Model and its applications: https://www.researchgate.net/publication/354201473_REVIEW_OF_THE_SPIRAL_MODEL_AND_ITS_APPLICATIONS
- dotnet-bot.com. (2025). *SafeBuffer Class*. dotnet-bot.com: <https://learn.microsoft.com/en-us/dotnet/api/system.runtime.interopservices.safebuffer?view=net-9.0>
- ECOticias. (2022, junio 16). *Quema de desechos sólidos en Latinoamérica*. Noticias de Ecología y Medio Ambiente: <https://www.ecoticias.com/desechos-reciclaje/quema-dedesechos-solidos-en-latinoamerica>
- Enríquez, E. (2009). *Racionalización de rutas y frecuencias del transporte terrestre urbano de la ciudad de Tulcán, provincia del Carchi*. Tesis de maestría, Escuela Politécnica Nacional. <https://doi.org/http://bibdigital.epn.edu.ec/handle/15000/8412>
- Fierro, E. (2023). *El ecosistema de Javascript en 2023*. *eduardofierro.com*: <https://eduardofierro.pro/blog/el-ecosistema-de-javascript-en-2023/>
- Figueroa, R., Solís, C., & Cabrera, A. (2016). *Universidad Técnica Particular de Loja*. <https://doi.org/10.13140>
- Flórez, M. (2023, Noviembre 17). *SITCA*. Retrieved Abril 24, 2025, from ¿Qué es un sistema de planificación de rutas?: <https://www.sitcaglobal.com/blog/nwarticle/166/10/que-es-un-sistema-de-planificacion-de-rutas>
- Fuchs, M., & Ntokos, A. (21 de Noviembre de 2023). *Study.com*. Retrieved 24 de Abril de 2025, from Directed vs. Undirected Graphs | Overview, Examples & Algorithms: <https://study.com/academy/lesson/weighted-graphs-implementation-dijkstra-algorithm.html>
- GADMCM. (2025, Febrero 12). *Gob.ec*. Retrieved Abril 24, 2025, from Servicio de Recolección de Residuos Sólidos Comunes: <https://www.gob.ec/gadmcm/tramites/servicio-recoleccion-residuos-solidos-comunes>
- García, C. (n.d.). *Introducción a Grafos*. <https://doi.org/https://www.cosijopii.com/assets/pdf/GrafosED.pdf>
- García, F. (03 de Agosto de 2023). (Arsys, Editor) Retrieved 24 de Abril de 2025, from Metodología DevOps: qué es y para qué sirve: <https://www.arsys.es/blog/metodologia-devops>
- García, L. (2024, noviembre 21). *5 librerías de Python para trabajar con OpenStreetMap*. MappingGIS: <https://mappinggis.com/tag/openstreetmap/>

- García, O. (2013). *Modelo de prototipos*. Proyectum. <https://doi.org/https://proyectum.com/sistema/blog/modelo-de-prototipos/>
- García, O. (02 de Septiembre de 2013). *Proyectum*. Retrieved 24 de Abril de 2025, from Modelo de prototipos: <https://proyectum.com/sistema/blog/modelo-de-prototipos/#:~:text=El%20modelo%20de%20prototipos%2C%20tambi%C3%A9n,lo%20que%20ser%C3%A1%20el%20sistema.>
- Ghosh, D., Das, K., Halder, P., & Sen, A. (2024, Agosto 20). *Physical Review Journals*. Retrieved Abril 24, 2025, from Entanglement of weighted graphs uncovering transitions in variable-range interacting models: <https://journals.aps.org/prj/abstract/10.1103/PhysRevA.110.022431>
- GitHub.io. (2025). *MySQL2*. Github.io: <https://sidorares.github.io/node-mysql2/docs>
- GoDaddy. (2024). *Flexbox: Guía para conseguir un diseño web flexible y adaptable*. GoDaddy. <https://doi.org/https://www.godaddy.com/resources/latam/desarrollo/flexbox-css-herramienta-esencial-maquetacion-web>
- Gómez, S. (2019, Mayo 23). *Universidad Nacional del Sur*. Retrieved Abril 24, 2025, from Estructuras de Datos: <https://cs.uns.edu.ar/~mlg/ed/downloads/Teorias/uns-ed-2019-clase-15-Grafos-1x2.pdf>
- Gonzales, D. (2023, junio 9). *¿Por qué los desarrolladores adoran Axios?* metstudio.com: <https://www.meltstudio.co/post/por-que-los-desarrolladores-adoran-axios>
- GoRetro Team. (2022, agosto 22). *Is Scrum applicable to a small organization?* GoRetro: <https://www.goretro.ai/answers/is-scrum-applicable-to-a-small-organization>
- Graph Everywhere. (2019, agosto 5). *Grafos | Qué son, tipos, orden y herramientas de visualización*. <https://www.grapheverywhere.com/grafos-que-son-tipos-orden-y-herramientas-de-visualizacion/>
- Graph Everywhere. (2024, Octubre 17). *Conectamos los Datos de las empresas mediante Grafos*. Graph Everywhere: <https://www.grapheverywhere.com/>
- greekforgeeks.org. (2020, julio 22). *Node.js util.deprecate() Method*. greekforgeeks.org: <https://www.geeksforgeeks.org/node-js-util-deprecate-method/>
- Grinberg, D. (2022, Abril 26). *Drexel University*. Retrieved Abril 24, 2025, from An introduction to Graph Theory: <https://www.cip.ifi.lmu.de/~grinberg/t/22s/graphs.pdf>
- Grupo Sulo. (2021, diciembre 13). *Sulo*. Retrieved Abril 29, 2025, from Optimización en el sistema Integral de gestión de la Pre-recogida: <https://mx.sulo.com/sistemas-de-gestion-de-la-pre-recogida/>

- Guachamin, G., & Alvarez, I. (2022). DESARROLLO DE APLICACIÓN WEB Y MÓVIL PARA PROVEER INFORMACIÓN DE LA RECOLECCIÓN Y MANEJO DE DESECHOS EN QUITO . ESCUELA DE FORMACIÓN DE TECNÓLOGOS.
- Gustavo, B. (2024, septiembre 3). ¿Qué es MySQL? Retrieved Abril 26, 2025, from Hostinger Tutoriales: <https://www.hostinger.es/tutoriales/que-es-mysql>
- Gutiérrez, M., & Pérez, L. (2019). *Tecnologías móviles y sistemas de posicionamiento*. Alfaomega.
- Hayes, M., & Downie, A. (5 de Junio de 2024). IBM. Retrieved 24 de Abril de 2025, from ¿Qué es Oracle?: <https://www.ibm.com/mx-es/topics/oracle>
- Hernández, R. (2014). *Metodología de la investigación* (Sexta ed.). McGraw-Hill Educatio. <https://www.esup.edu.pe/wp-content/uploads/2020/12/2.%20Hernandez,%20Fernandez%20y%20Baptista-Metodolog%C3%ADa%20Investigacion%20Cientifica%206ta%20ed.pdf>
- llimit. (2022, marzo 18). *Metodología SCRUM: qué es y cómo implementarlo*. llimit: <https://llimit.com/blog/metodologia-scrum/>
- Imaginaformacion.com. (2024, octubre 17). ¿Qué es un Scrum Team? Roles y Funciones. <https://imaginaformacion.com/tutoriales/que-es-un-scrum-team>
- Incinerox. (2018, Abril 12). *Incinerox*. Retrieved Abril 24, 2025, from ¿Cuál es el proceso adecuado para el manejo de residuos sólidos?: <https://incinerox.com.ec/cuales-el-proceso-adecuado-para-el-manejo-de-residuos-solidos/>
- Instituto tecnológico de Nuevo Laredo. (2005, Abril 24). *Algoritmo A**. Retrieved 2025, from Instituto tecnológico de Nuevo Laredo: [https://nlaredo.tecnm.mx/takeyas/apuntes/Inteligencia%20Artificial/Apuntes/tareas_alumnos/A-Star/A-Star\(2005-II-A\).pdf](https://nlaredo.tecnm.mx/takeyas/apuntes/Inteligencia%20Artificial/Apuntes/tareas_alumnos/A-Star/A-Star(2005-II-A).pdf)
- Instituto tecnológico de Nuevo Laredo. (2005). *Algoritmo A**. Instituto tecnológico de Nuevo Laredo: [https://nlaredo.tecnm.mx/takeyas/apuntes/Inteligencia%20Artificial/Apuntes/tareas_alumnos/A-Star/A-Star\(2005-II-A\).pdf](https://nlaredo.tecnm.mx/takeyas/apuntes/Inteligencia%20Artificial/Apuntes/tareas_alumnos/A-Star/A-Star(2005-II-A).pdf)
- Jaulent, E. (2024). *Metodología RAD*. UCloud. <https://doi.org/https://ucloudglobal.com/blog/metodologia-rad>
- Javaid, A. (2013). *Understanding Dijkstra Algorithm*. ResearchGate: https://www.researchgate.net/publication/273264449_Understanding_Dijkstra_Algorithm
- Jesús. (2022). ¿Qué es Node JS? Dongee. <https://doi.org/https://www.dongee.com/tutoriales/que-es-node-js/>
- Juvencio, G., & Domínguez, V. (2022, diciembre 13). *El manejo de los desechos sólidos en la parroquia Chanduy, Cantón Santa Elena*. Repositorio UPSE:

chrome-
extension://efaidnbmnnnibpcajpcglclefindmkaj/https://repositorio.upse.edu.
ec/xmlui/bitstream/handle/46000/3916/UPSE-TOD-2017-
0013.pdf?sequence=1&isAllowed=y

Keepcoding.io. (2022, abril 11). *¿Qué es MySQL Workbench?* Keepcoding.io:
<https://keepcoding.io/blog/que-es-mysql-workbench/>

kinsta.com. (2022, diciembre 19). *¿Qué es Express.js? Todo lo que Debes Saber.*
kinsta.com: <https://kinsta.com/es/base-de-conocimiento/que-es-express/>

Kumar, D., & Misra, A. (2016). *Graph theory with applications*. Springer.
<https://doi.org/10.1007/978-81-322-3685-4>

Laoyan, S. (2025, Febrero 6). *Asana*. Retrieved Abril 25, 2025, from *Qué es la metodología waterfall y cuándo utilizarla:*
<https://asana.com/es/resources/waterfall-project-management-methodology>

Layout, C. G. (2024). *Diseño de cuadrícula CSS*. MDN WEB DOCS.
[https://doi.org/https://developer.mozilla.org/es/docs/Web/CSS/CSS_grid_layo
u](https://doi.org/https://developer.mozilla.org/es/docs/Web/CSS/CSS_grid_layout)

learn.microsoft.com. (2024, octubre 10). *Azure Core Util client library for JavaScript - version 1.11.0*. learn.microsoft.com: [https://learn.microsoft.com/en-
us/javascript/api/overview/azure/core-util-readme?view=azure-node-latest](https://learn.microsoft.com/en-us/javascript/api/overview/azure/core-util-readme?view=azure-node-latest)

learn.microsoft.com. (2025). *SqlString Estructura*. learn.microsoft.com:
[https://learn.microsoft.com/es-
es/dotnet/api/system.data.sqltypes.sqlstring?view=net-8.0](https://learn.microsoft.com/es-es/dotnet/api/system.data.sqltypes.sqlstring?view=net-8.0)

Leite, R. A., & Souza, R. A. (2021). Dynamic models for planning and scheduling the collection of solid waste. *El Sevier*, 121, 186-197. <https://doi.org/10/1016>

León, A., Acosta, J., & Díaz, R. (2 de Octubre de 2021). *SciELO Cuba*. Retrieved 24 de Abril de 2025, from *Aplicación de la metodología incremental en el desarrollo de sistemas de información:*
[http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2218-
36202021000500175](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2218-36202021000500175)

Leremenko, D. (2020, Marzo 12). *MINDK*. Retrieved Abril 24, 2025, from *ArangoDB: Perfect Database for Projects with High Uncertainty:*
<https://www.mindk.com/blog/arangodb/>

Liu, Y., Dong, H., Lohse, n., & Petrovic, S. (2016). A multi-objective genetic algorithm for optimisation of energy consumption and shop floor production performance. *Elsevier*, 179, 259-272. <https://doi.org/10.1016/j.jipe>

logsnag.com. (2023, octubre 20). *How to Handle Time-Zones in JavaScript with date-fns-tz*. logsnag.com: [https://logsnag.com/blog/handling-timezones-in-
javascript-with-date-fns-tz](https://logsnag.com/blog/handling-timezones-in-javascript-with-date-fns-tz)

- Lomelí, L. (26 de mayo de 2023). *Innevo*. Retrieved 24 de 04 de 2015, from Metodología Scrum: Roles, Procesos y Artefactos: <https://innevo.com/blog/metodologia-scrum>
- Lomelí, L. (2023, mayo 26). *Metodología Scrum: Roles, Procesos y Artefactos*. Innevo.com: <https://blog.innevo.com/metodologia-scrum>
- López, M., Villagra, A., & Pandolfi, D. (7 de Febrero de 2023). *Informes CT-UNPA*. <https://doi.org/10.26457>
- Loza, S. (2011, diciembre 2). *Algoritmo de Dijkstra y Algoritmo de Bellman-Ford*. Prezi: <https://prezi.com/7egkjhlo-mgl/algoritmo-de-dijkstra-y-algoritmo-de-bellman-ford/>
- Lucano Chávez, F. B. (2019). *Optimización de rutas para la recolección de desechos sólidos en la ciudad de Ibarra provincia de Imbabura*. Universidad Técnica del Norte. <https://doi.org/https://repositorio.utn.edu.ec/handle/123456789/9096>
- Lucano, C. (2019, abril 25). *Optimización de rutas para la recolección de desechos sólidos en la ciudad de Ibarra provincia de Imbabura*. Repositorio Digital Técnica del Norte: <https://repositorio.utn.edu.ec/handle/123456789/9096>
- Lucano, F. (25 de Abril de 2019). *Universidad Técnica del Norte*. Retrieved 24 de Abril de 2025, from Optimización de rutas para la recolección de desechos sólidos en la ciudad de Ibarra provincia de Imbabura: <https://repositorio.utn.edu.ec/handle/123456789/9096>
- Mahajan, R. (2023). Environment and Health Impact of Solid Waste Management in Developing Countries: A Review. *ResearchGate*, 3(18), 18-29. <https://doi.org/10/12944>
- Marchena, P. (2015). Desarrollo de un sistema para la optimización de rutas de trabajo utilizando el algoritmo de Dijkstra y diagramas de Voronoi. *Universidad de Sipán*, 3. <https://doi.org/20.500.12802/157>
- Márquez , F. J., Rodríguez, K., & Garro, B. A. (2020). Diseño de rutas de recolección basura utilizando el algoritmo de optimización por colonia de hormigas. *Revista Del Centro de Investigación de La Universidad La Salle*, 13(52), 19-66. <https://doi.org/10.26457/recein.v13i51.1925>
- Márquez, F., Rodríguez, K., & Garro, B. (03 de Junio de 2020). *Universidad La Salle*. <https://doi.org/10.26457>
- Martins, J. (2025, Febrero 15). *Asana*. Retrieved 04 24, 2025, from Scrum: conceptos clave y cómo se aplica en la gestión de proyectos: <https://asana.com/es/resources/what-is-scrum>
- Martins, J. (2025, Enero 19). *Asana*. Retrieved Abril 24, 2024, from ¿Qué es la metodología Kanban y cómo funciona?: <https://asana.com/es/resources/what-is-kanba>

- Mata, L. (2019, mayo 21). *El enfoque cuantitativo de investigación*. Investigalia: <https://investigaliacr.com/investigacion/el-enfoque-cuantitativo-de-investigacion/>
- Mata, S. (2019, mayo 28). *El enfoque cualitativo de investigación*. Investigalia: <https://investigaliacr.com/investigacion/el-enfoque-cualitativo-de-investigacion/>
- Mathworks. (2021). *Graph and Network Algorithms*. <https://www.mathworks.com/help/matlab/graph-and-network-algorithms.html>
- MDN Web Docs. (2024, diciembre 17). *Array.isArray()*. MDN Web Docs: https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Array/isArray
- MDN Web Docs. (2024, agosto 30). *ReadableStream*. MDN Web Docs: <https://developer.mozilla.org/en-US/docs/Web/API/ReadableStream>
- Microsoft. (2021, Noviembre 3). *Visual Studio Code*. VisualStudio.com: <https://code.visualstudio.com/docs/editor/whyvscode>
- Microsoft. (2022). *Visual Studio 2022*. Microsoft: <https://visualstudio.microsoft.com/es/vs/>
- Ministerio de Inclusión Económica y Social. (2024). *Ejecutar programas y servicios para la*. Inclusion.gob.ec. : <https://www.inclusion.gob.ec>.
- MITECO. (2024). *Sistema de Recogida*. Retrieved Abril 24, 2025, from [miteco.go.es: https://www.miteco.gob.es/es/calidad-y-evaluacion-ambiental/temas/prevencion-y-gestion-residuos/flujos/domesticos/gestion/sistema-recogida/default_antigua.html](https://www.miteco.gob.es/es/calidad-y-evaluacion-ambiental/temas/prevencion-y-gestion-residuos/flujos/domesticos/gestion/sistema-recogida/default_antigua.html)
- Molina, H. (2024, Octubre 17). *Innovación Pedagógica*. Retrieved Abril 24, 2025, from Scrum Educativo: Impulsando el trabajo colaborativo: <https://ucontinental.edu.pe/innovacionpedagogica/scrum-educativo-impulsando-el-trabajo-colaborativo/notas-destacadas/>
- Mongodb.com. (2018). *¿Qué es MongoDB?* MongoDB: <https://www.mongodb.com/es/company/what-is-mongodb>
- Morán, S. (2018, julio 23). *Plan V*. <https://planv.com.ec/historias/sociedad/basura-numeros-rojos-ecuador/>
- Mukhlif, F., & Saif, A. (2020, febrero 2). *Comparative Study On Bellman-Ford And Dijkstra Algorithms*. Retrieved Abril 29, 2025, from ResearchGate: https://www.researchgate.net/publication/340790429_Comparative_Study_On_Bellman-Ford_And_Dijkstra_Algorithms

- Mysql. (2021, octubre 21). *Seguridad en MySQL*. <https://www.mysql.com/news-and-events/web-seminars/security-in-mysql-es-2021>
- Nan, Z., Fang, Q., Bang, L., & Hong, G. (2021). Comparing Machine-Learning Models for Drought Forecasting in Vietnam's Cai River Basin. *Environment Study*, 27(80866), 2633-2646. <https://doi.org/10.15244>
- Narvaez, M. (2022, Octubre 3). *QuestionPro*. Retrieved Abril 24, 2025, from Método deductivo: Qué es y cuál es su importancia: <https://www.questionpro.com/blog/es/metodo-deductivo/>
- Narvaez, M. (2024, Mayo 9). *QuestionPro*. Retrieved Abril 24, 2025, from Método inductivo: Qué es, características y ejemplos: <https://www.questionpro.com/blog/es/metodo-inductivo>
- neo4j.com. (2024). *Neo4j Drivers*. neo4j.com: <https://neo4j.com/docs/bolt/current/neo4j-drivers/>
- Network 360. (2023, mayo 5). *MySQL Workbench: Qué es, descarga, instalación y uso*. <https://www.innovaciondigital360.com/big-data/guia-completa-de-mysql-workbench-descarga-instalacion-y-uso/>
- nodejs.org. (2025). *Node.js v23.6.1 documentation*. nodejs.org: https://nodejs.org/api/string_decoder.html
- nodemon.io. (2025). *Nodemon*. nodemon.io: <https://nodemon.io/>
- Nomadia. (2023, octubre 2023). *Sistema de Geolocalización en la Empresa: Usos Apropriados*. Nomandia: <https://www.nomadia-group.com/es/recursos/blog/sistema-de-geolocalizacion-en-la-empresa-usos-apropiados/>
- npmjs.com. (2024, septiembre 30). *date-fns-tz*. npmjs.com: <https://www.npmjs.com/package/date-fns-tz>
- Obando, J. (2022, Octubre 28). *REpositorio Digital Universidad Técnica del Norte*. Retrieved Abril 24, 2025, from Sistema de recolección de desechos optimizado, mediante IOT en el sector El Olivo de la ciudad de Ibarra provincia de Imbabura: <https://repositorio.utn.edu.ec/handle/123456789/13118>
- Olawanle, J. (2023, julio 13). *Las 30 Mejores extensiones de VS Code para Mejorar la experiencia de programación*. Kinsta: <https://kinsta.com/es/blog/extensiones-vscode/>
- OpenStreetMap. (2025). *OpenStreetMap*. <https://www.openstreetmap.org/#map=14/0.80910/-77.71523>
- openwebinars.net. (2019, septiembre 24). *Qué es MySQL: Características y ventajas*. openwebinars.net: <https://openwebinars.net/blog/que-es-mysql/>

- Oracle.com. (2021). *Definición de base de datos orientada a grafos*. Oracle.com: <https://www.oracle.com/mx/autonomous-database/what-is-graph-database/>
- Ortega, L. (3 de Enero de 2022). *Lean Management*. Retrieved 24 de Abril de 2025, from Metodología RUP: ¿Qué es, cuál es su objetivo y cómo se utiliza?: <https://lean-management.site/rup/>
- Ortiz, R., & Gonzáles, D. (2019). *Gestión de operaciones logísticas*. Alfaomega. <https://doi.org/9786075388012>
- Ortuño, M. (2023, Diciembre). *Universidad Rey Juan Carlos*. Retrieved Abril 24, 2025, from Express: <https://gsyc.urjc.es/~mortuno/at/express.pdf>
- Prefectura del Carchi. (2024). *Prefectura del Carchi*. Dirección de Gestión Ambiental: https://carchi.gob.ec/gob_electronico/
- Primefaces. (2025). *PrimeFlex*. <https://www.primefaces.org/showcase/primeflex/setup.xhtml?jfwid=be3f8>
- Primevue. (2025). *Primeflex*. <https://primevue.org/>
- Putri, A. (2024, Junio). *ResearchGate*. Retrieved Abril 24, 2024, from Graph Theory: Application of graphs to the shortest path problem with Dijkstra's algorithm: https://www.researchgate.net/publication/381508818_Graph_Theory_Application_of_graphs_to_the_shortest_path_problem_with_Dijkstra's_algorithm
- Pykes, K. (2024, Abril 27). *Datacamp*. Retrieved Abril 24, 2025, from ¿Qué es una base de datos de grafos?: <https://www.datacamp.com/es/blog/what-is-a-graph-database>
- Quintana, E., Ojeda, L., & Almeida, O. (18 de Septiembre de 2019). Diseño de un sistema de rutas variable de transportación basado en sistemas de información geográfica. *Instituto de Información Científica y Tecnológica, XXI(4)*, 512-530. Retrieved 24 de Abril de 2025, from <https://www.redalyc.org/journal/6378/637869114009/html/>
- Radigan, D. (2025). *Product Backlog - What is it & How to create one*. Atlassian: <https://www.atlassian.com/agile/scrum/backlogs#:~:text=What%20is%20a%20product%20backlog,knowns%20what%20to%20deliver%20first>.
- Radkovskyy, B., & Muzyka, R. (2024, Enero 2). *Leobit*. Retrieved Abril 24, 2024, from Why to Use Angular in 2024: An Overview of Angular Pros and Cons: <https://leobit.com/blog/why-to-use-angular-in-an-overview-of-angular-pros-and-cons/>
- Raeburn, A. (2024, febrero 2). *¿Qué es un Scrum Master y cuál es su función?* Asana: <https://asana.com/es/resources/scrum-master>

- Ravikiran, A. (2025, Febrero 25). *Simplilearn*. Retrieved Abril 24, 2025, from Graph in Data Structure | Types & Explanation: <https://www.simplilearn.com/tutorials/data-structure-tutorial/graphs-in-data-structure>
- Reaiche, C. (5 de Agosto de 2022). *James Cook University*. Retrieved 24 de Abril de 2025, from The traditional, sequential methodologies: <https://jcu.pressbooks.pub/pmmethods/chapter/traditional-sequential-methodologies/#:~:text=Traditional%20or%20sequential%20project%20management,%3B%20Jovanovic%20and%20Beric%202018>).
- Repsol. (2024). *Metodología agile*. Repsol. <https://doi.org/https://www.repsol.com/es/energia-futuro/personas/metodologia-agile/index.cshtml>
- Reyna, S. (2021, Mayo 13). *Seredecom*. Retrieved Abril 13, 2025, from ¿En qué consiste el servicio de recolección de basura?: <https://www.recolecciondebasuraseredecom.com.mx/en-que-consiste-el-servicio-de-recoleccion-de-basura>
- Robledano, A. (2019). *Qué es MySQL: Características y ventajas*. OpenWebinars. <https://doi.org/https://openwebinars.net/blog/que-es-mysql/>
- Rodas, G. (2019, agosto 21). *¿Qué es Boom en Hapi JS?* Platzy.com: <https://platzi.com/clases/1646-backend-nodejs-2019/22041-que-es-joi-y-boom/#:~:text=%C2%BFQu%C3%A9%20es%20Boom%20en%20Hapi,los%20debemos%20hacer%20es%3A%20Boom>.
- Rodón, A. (1 de Julio de 2022). *En mi local funciona*. Retrieved 24 de Abril de 2025, from ¿Conoces Neo4j o sabes de qué va?: <https://www.enmilocalfunciona.io/conoces-neo4j-o-sabes-de-que-va/>
- Román, C. (27 de febrero de 2025). *Monolitic*. Retrieved 24 de Abril de 2025, from Tecnologías de geolocalización para entornos urbanos: <https://www.monolitic.com/tecnologias-de-geolocalizacion-de-altas-prestaciones-para-entornos-urbanos/>
- Romero, J., Obregón, S., Quezada, E., & Torres, M. (2016). *Evaluación de rutas de recolección de residuos sólidos urbanos con apoyo de dispositivos de rastreo satelital: análisis e implicaciones*. SCIELO. <https://doi.org/https://doi.org/10.20937/RICA.2016.32.03.07>
- Rondón, A. (2022, julio). *¿Conoces Neo4j o sabes de qué va?* En mi local funciona: <https://www.enmilocalfunciona.io/conoces-neo4j-o-sabes-de-que-va/>
- rpubs.com. (2017, junio 6). *Librería XLSX en R*. rpubs.com: <https://rpubs.com/ADAUTOR/282650>
- Ruedas, L., & Mendoza, I. (2021, Enero 31). *Medium*. Retrieved Abril 29, 2025, from Algoritmo de Floyd-Warshall: <https://medium.com/algoritmo-floyd-warshall/algoritmo-de-floyd-warshall-e1fd1a900d8>

- Russell, S., & Norving, P. (2010). *Artificial intelligence: a modern approach*. Prentice Hall, 4.5(3).
- Sáez, J. (2021, diciembre 3). *Cómo funciona la Metodología Scrum: Qué es y cómo utilizarla*. IEBS: <https://www.iebschool.com/blog/metodologia-scrum-agile-scrum/>
- Salazar, B. (2019, junio 12). *Algoritmo de Dijkstra*. Retrieved Abril 24, 2025, from Ingeniería Industrial Online: <https://ingenieriaindustrialonline.com/investigacion-de-operaciones/algoritmo-de-dijkstra/>
- Sanchez, J. (2021). *Universidad de los Andes Colombia*. Retrieved 2024 de Abril de 2025, from Diseño de rutas para la recolección de residuos sólidos implementando herramientas computacionales del sistema de información geográfica (SIG) ArcGIS en un municipio colombiano.: <https://repositorio.uniandes.edu.co/entities/publication/8c207ddd-6b82-42c0-a795-e0b16616da4d>
- Santillán, C. (2019). *Tendencias futuras aplicadas al transporte y movilidad sostenible*. Escuela Superior Politécnica de Chimborazo. <https://doi.org/http://cimogsys.esepoch.edu.ec/direccion-publicaciones/public/docs/books/2020-01-28-143846-Libro%20Memorias%20Coniadt%202019.pdf>
- Saramaniego, G. (28 de Agosto de 2022). *Mi asesor de tesis*. Retrieved 24 de Abril de 2025, from Enfoque, tipo, diseño y método de investigación: https://miasesorde tesis.com/enfoque-tipo-diseno-metodo-de-investigacion/#Enfoque_cuantitativo
- Sarker, A., Billah, M., Islam, N., & Kumar, A. (15 de Julio de 2021). COVID-19 pandemic and healthcare solid waste management strategy – A mini-review. *ELSEVIER*, 778, 892. <https://doi.org/10.1016>
- Sarkkinen, M., Kuajala, K., & Seppo, G. (2019). Decision support framework for solid waste management based on . *Science Direct*, 51, 246-258. <https://doi.org/10.1016>
- Science Direct. (2023). *Directed Graphs*. <https://www.sciencedirect.com/topics/computer-science/directed-graphs>
- Sclar, M. (2015). *Universidad de Buenos Aires*. Retrieved Abril 24, 2025, from <https://www.oia.unsam.edu.ar/wp-content/uploads/2013/10/Grafos.pdf>
- Scrum.org. (2025). *What is Scrum?* Scrum.org: <https://www.scrum.org/>
- Scrumio.com. (2021, febrero 7). *El Incremento de producto*. Scrumio.com: <https://www.scrumio.com/scrum/incremento-de-producto/>

- Sebesta, R. (2012). *Concepts of Programming Languages (10th ed.)*. Colorado: Pearson. Retrieved Abril 24, 2025, from <https://www.ime.usp.br/~alvaroma/ucsp/proclang/book.pdf>
- Simic, M. (2024, Marzo 18). *Baeldung*. Retrieved Abril 29, 2025, from *Weighted vs. Unweighted Graphs*: <https://www.baeldung.com/cs/weighted-vs-unweighted-graphs>
- skysnag.com. (2023, octubre 11). *¿Qué es bcrypt?* skysnag.com: <https://www.skysnag.com/es/blog/what-is-bcrypt/>
- Smartup, A. (2024, Diciembre 3). *Arangoya*. Retrieved Abril 24, 2025, from *Base de datos relacional vs no relacional: diferencias y usos*: <https://arangoya.org/base-de-datos-relacional-no-relacional/#:~:text=Si%20tu%20prioridad%20es%20la,opta%20por%20una%20no%20relacional>
- Solbyte. (2023, junio 15). *Node JS: ventajas e inconvenientes*. <https://www.solbyte.com/blog/nodejs/>
- Sommerville, I. (2011). *Ingeniería del software (9.ª ed.)*. Pearson Educación.
- somospnt.com. (2020, febrero 5). *Flexbox - ¿Qué es y para qué sirve?* somospnt.com: <https://somospnt.com/blog/148-flexbox-que-es-y-para-que-sirve>
- stackoverflow.com. (2020, septiembre 25). *Qué es Body-parse y para qué sirve*. stackoverflow.com: <https://es.stackoverflow.com/questions/392306/qu%C3%A9-es-body-parse-y-para-qu%C3%A9-sirve>
- Steinhardt, G. (29 de Mayo de 2024). *Medium*. Retrieved 24 de Abril de 2025, from *The MSF Team Model and Product Management*: <https://medium.com/blackblot/the-msf-team-model-and-product-management-9f3411e355c5>
- Stride. (2023, enero 13). *Javascript: ¿Qué es y para qué sirve?* Stride: <https://stride.com.co/blog/javascript-que-es-para-que-sirve/>
- Svelte.dev. (2025). *Svelte.dev*. Retrieved Abril 24, 2025, from *Svelte.dev*: <https://svelte.dev/>
- Talha, S., Rizwan, S., & Rizwan, S. (Diciembre de 2024). *ResearchGate*. Retrieved 24 de Abril de 2025, from *Optimal Route Assessment of Oil and Gas Pipelines using Geographic Information System (GIS)*: https://www.researchgate.net/publication/386989248_Optimal_Route_Assessment_of_Oil_and_Gas_Pipelines_using_Geographic_Information_System_GIS
- Tamushi. (18 de Mayo de 2022). *Testing iT*. Retrieved 24 de Abril de 2025, from *4 normas ISO que definen estándares de calidad de software*: <https://www.testingit.com.mx/blog/estandares-calidad-software>

- Tang, J., & Qingtao, P. (2021). A review of swarm intelligence algorithms for solving optimization problems: applications and trends. *Journal of Automatica Sinica*, 1643(10), 1627. <https://doi.org/10.1016>
- Team Asana. (2024, febrero 6). *Backlog: qué es el trabajo pendiente del sprint y ejemplos*. Team Asana: <https://asana.com/es/resources/sprint-backlog>
- Thulasiraman, K., & Swamy, M. (2016). *Graphs: Theory and algorithms*. John Wiley & Sons. <https://doi.org/10.1002/9781119253084>
- Tokio School. (2021, mayo 25). *¿Qué es la geolocalización en aplicaciones?* . Tokio School: <https://www.tokioschool.com/noticias/que-es-geolocalizacionuso-aplicaciones/>
- Toro, E., Szanto, M., Juan , N., Pacheco, F., Contreras, E., & Gálvez, A. (n.d.). *Guía general para la gestión de desechos sólidos domiciliarios*. <https://repositorio.cepal.org/login>
- Turton, P. (2024, Noviembre 7). *Viewpoint Analysis*. Retrieved Abril 24, 2025, from What is Route Planning Software and Who Uses It?: <https://www.viewpointanalysis.com/post/what-is-route-planning-software-and-who-uses-it>
- Universal Service Administrative Co. (2025). *Geolocation Methods*. <https://www.usac.org/wp-content/uploads/high-cost/documents/Tools/HUBBGeolocationMethods.pdf>
- Villamarín, A., & Ortega, S. (2020). *Innovación tecnológica en la gestión de residuos sólidos urbanos*. *Revista Latinoamericana de Ingeniería Ambiental*. [https://doi.org/12\(2\), 97-107](https://doi.org/12(2), 97-107)
- Villarreal, E., & Edwin, R. (2009). *Racionalización de rutas y frecuencias del transporte terrestre urbano de la ciudad de Tulcán, provincia del Carchi [Tesis de maestría*. Escuela Politécnica Nacional. <https://doi.org/http://bibdigital.epn.edu.ec/handle/15000/8412>
- Villemain, C. (2018, octubre 12). *Cómo la basura afecta al desarrollo de América Latina*. Noticias ONU: <https://news.un.org/es/story/2018/10/1443562>
- Vue.js. (2025). *Vue.js*. Retrieved Abril 24, 2025, from ¿Qué es Vue.js?: <https://es.vuejs.org/v2/guide/>
- w3schools.com. (2025). *Java Inheritance (Subclass and Superclass)*. w3schools.com : https://www.w3schools.com/java/java_inheritance.asp
- Wattles, J. (2018, enero 19). *Coke: We'll recycle one can or bottle for every one we*. CNNMoney: <https://money.cnn.com/2018/01/19/news/companies/coke-coca-cola-zero-waste/index.html>

- Webber, J., & Eifrem, E. (2015). *Graph Databases: New Opportunities for Connected Data* (2nd ed. O'Reilly Media).
- WESchools. (2025). *CSS Grid Layout*. https://www.w3schools.com/css/css_grid.asp
- West, D. (2024). *Planificación de sprints*. Atlassian: <https://www.atlassian.com/es/agile/scrum/sprint-planning>
- Weyenberg, G., & Yoshida, R. (2015). Reconstructing the Phylogeny: Computational Methods. In *Algebraic and Discrete Mathematical Methods for Modern Biology* (pp. 293-319). Academic Press. Retrieved April 29, 2025
- Wikipedia.org. (2008, julio 22). *Matriz de distancias*. Wikimedia Foundation: https://es.wikipedia.org/wiki/Matriz_de_distancias
- Wilson, R. (1996). *Graph Theory*. Prentice Hall. <https://doi.org/0-582-24993-7>
- Woods, C. (2023, Mayo 23). *Locate2u*. Retrieved April 24, 2025, from Mastering Route Management in 2023: A Comprehensive Guide: <https://www.locate2u.com/route-planning/mastering-route-management-in-2023-a-comprehensive-guide/>
- World Bank. (2018, septiembre 20). *Grupo Banco Mundial*. Los desechos: un análisis actualizado del futuro de la gestión de los desechos sólidos: <https://www.bancomundial.org/es/news/immersive-story/2018/09/20/what-a-waste-an-updated-look-into-the-future-of-solid-waste-management>
- Yungon, C., Salazar, E., & Vilacrés, J. (2022, julio 19). *Algoritmo de Bellman Ford para solucionar el problema de la ruta más corta entre nodos*. <https://polodelconocimiento.com/ojs/index.php/es/article/view/4285/html>
- Zambrano, A. (2021, marzo 12). *Modelo predictivo para la toma de decisiones en la*. Monografias.com: <https://www.monografias.com/docs110/modelo-predictivo-toma-decisiones-gestion-insumos-y-medicamentos/modelo-predictivo-toma-decisiones-gestion-insumos-y-medicamentos>
- Zambrano, R. (2020, diciembre 30). *Ecuador genera 375 mil toneladas de desechos sólidos urbanos al año, pero solo recicla el 4% de estos desechos*. Eluniverso.com: <https://www.eluniverso.com/noticias/2020/12/30/nota/9111586/ecuador-genera-375-mil-toneladas-desechos-solidos-urbanos-ano-solo/>
- Zheng, Y., Wang, W., Chen, Y., & Wu, D. (2019). A new Dijkstra algorithm based on improved heap data structure. *IEEE Access*, 7, 8260-8271. <https://doi.org/10.1109/ACCESS.2019>
- Zhu, L., Fei, R. Y., Wang, Y., & Ning, B. (2018). Big Data Analytics in Intelligent Transportation Systems: A Survey. *ResearchGate*, 99, 1-16. <https://doi.org/10.1109>

VII. ANEXOS

Anexo 1. Acta de la sustentación de Predefensa del TIC

UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI

FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES

CARRERA DE COMPUTACIÓN

ACTA

DE LA SUSTENTACIÓN ORAL DE LA PREDEFENSA DEL TRABAJO DE INTEGRACIÓN CURRICULAR

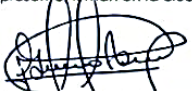
ESTUDIANTE:	CADENA CASTRO ESTEFANÍA LIZBETH	CÉDULA DE IDENTIDAD:	0401992268
PERIODO ACADÉMICO:	2025A		
PRESIDENTE TRIBUNAL	MSC. Georgina Guadalupe Arcos Ponce	DOCENTE TUTOR:	MSC. Jairo Vladimir Hidalgo Gujano
DOCENTE:	MSC. Samuel Benjamín Lascano Rivera		
TEMA DEL TIC:	"Sistema de gestión de rutas para la recolección de desechos sólidos"		


No.	CATEGORÍA	Evaluación cuantitativa	OBSERVACIONES Y RECOMENDACIONES
1	PROBLEMA - OBJETIVOS	10.00	
2	FUNDAMENTACIÓN TEÓRICA	10.00	
3	METODOLOGÍA	10.00	
4	RESULTADOS	10.00	
5	DISCUSIÓN	10.00	
6	CONCLUSIONES Y RECOMENDACIONES	10.00	
7	DEFENSA, ARGUMENTACIÓN Y VOCABULARIO PROFESIONAL	10.00	Controlar los nervios, mirar al tribunal.
8	FORMATO, ORGANIZACIÓN Y CALIDAD DE LA INFORMACIÓN	8.67	Revisar formatos, normas APA, ortografía y redacción de todo el documento, revisar bibliografía, revisar formatos de fuentes


Obteniendo una nota de: **9,60** Por lo tanto, **APRUEBA** ; debiendo el o los investigadores acatar el siguiente artículo:

Art. 36.- De los estudiantes que aprueban el informe final del TIC con observaciones.- Los estudiantes tendrán el plazo de 10 días para proceder a corregir su informe final del TIC de conformidad a las observaciones y recomendaciones realizadas por los miembros del Tribunal de sustentación de la pre-defensa.

Para constancia del presente, firman en la ciudad de Tulcán el **21 de mayo de 2025**


MSC. Georgina Guadalupe Arcos Ponce
PRESIDENTE TRIBUNAL


MSC. Jairo Vladimir Hidalgo Gujano
DOCENTE TUTOR


MSC. Samuel Benjamín Lascano Rivera
DOCENTE



UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI



FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES

CARRERA DE COMPUTACIÓN

ACTA


DE LA SUSTENTACIÓN ORAL DE LA PREDEFENSA DEL TRABAJO DE INTEGRACIÓN CURRICULAR


ESTUDIANTE	LECHÓN GUATEMAL JORGE RUBÉN		CÉDULA DE IDENTIDAD:	1728049139
PERIODO ACADÉMICO:	2025A			
PRESIDENTE TRIBUNAL	MSC. Georgina Guadalupe Arcos Ponce		DOCENTE TUTOR:	MSC. Jairo Vladimír Hidalgo Gullama
DOCENTE:	MSC. Samuel Benjamín Lascana Rivera			
TEMA DEL TIC:	"Sistema de gestión de rutas para la recolección de desechos sólidos"			
No.	CATEGORÍA	Evaluación cuantitativa	OBSERVACIONES Y RECOMENDACIONES	
1	PROBLEMA - OBJETIVOS	10,00		
2	FUNDAMENTACIÓN TEÓRICA	10,00		
3	METODOLOGÍA	10,00		
4	RESULTADOS	10,00		
5	DISCUSIÓN	10,00		
6	CONCLUSIONES Y RECOMENDACIONES	10,00		
7	DEFENSA, ARGUMENTACIÓN Y VOCABULARIO PROFESIONAL	10,00	Controlar los nervios, mirar al tribunal.	
8	FORMATO, ORGANIZACIÓN Y CALIDAD DE LA INFORMACIÓN	8,67	Revisar formatos, normas APA, ortografía y redacción de todo el documento, revisar bibliografía, revisar formatos de fuentes	


Obteniendo una nota de: **9,60** Por lo tanto, **APRUEBA** : debiendo el o los investigadores acatar el siguiente artículo:

Art. 36.- De los estudiantes que aprueban el Informe final del TIC con observaciones.- Los estudiantes tendrán el plazo de 10 días para proceder a corregir su Informe final del TIC de conformidad a las observaciones y recomendaciones realizadas por los miembros del Tribunal de sustentación de la pre-defensa.

Para constancia del presente, firmas en la ciudad de Tulcán el 21 de mayo de 2025


MSC. Georgina Guadalupe Arcos Ponce
PRESIDENTE TRIBUNAL


MSC. Jairo Vladimír Hidalgo Gullama
DOCENTE TUTOR


MSC. Samuel Benjamín Lascana Rivera
DOCENTE

Anexo 2. Certificado del abstract por parte de idiomas



UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI FOREIGN AND
NATIVE LANGUAGES CENTER

ABSTRACT- EVALUATION SHEET				
NAME: Lizbeth Estefanía Cadena Castro y Jorge Rubén Lechón Guatemala				
DATE: Viernes, 6 de junio de 2025				
Topic: Sistema de gestión de rutas para la recolección de desechos sólidos.				
MARKS AWARDED		QUANTITATIVE AND QUALITATIVE		
VOCABULARY AND WORD USE	Use new learnt vocabulary and precise words related to the topic	Use a little new vocabulary and some appropriate words related to the topic	Use basic vocabulary and simplistic words related to the topic	Limited vocabulary and inadequate words related to the topic
	EXCELLENT: 2 <input checked="" type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
WRITING COHESION	Clear and logical progression of ideas and supporting paragraphs.	Adequate progression of ideas and supporting paragraphs.	Some progression of ideas and supporting paragraphs.	Inadequate ideas and supporting paragraphs.
	EXCELLENT: 2 <input type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
ARGUMENT	The message has been communicated very well and identify the type of text	The message has been communicated appropriately and identify the type of text	Some of the message has been communicated and the type of text is little confusing	The message hasn't been communicated and the type of text is inadequate
	EXCELLENT: 2 <input type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
CREATIVITY	Outstanding flow of ideas and events	Good flow of ideas and events	Average flow of ideas and events	Poor flow of ideas and events
	EXCELLENT: 2 <input type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
SCIENTIFIC SUSTAINABILITY	Reasonable, specific and supportable opinion or thesis statement	Minor errors when supporting the thesis statement	Some errors when supporting the thesis statement	Lots of errors when supporting the thesis statement
	EXCELLENT: 2 <input type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
TOTAL/AVERAGE	9 - 10: EXCELLENT 7 - 8,9: GOOD 5 - 6,9: AVERAGE 0 - 4,9: LIMITED	TOTAL 9		



UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI- FOREIGN AND NATIVE LANGUAGES CENTER

Informe sobre el Abstract de Artículo Científico o Investigación.

Autor: Lizbeth Estefanía Cadena Castro y Jorge Rubén Lechón Guatemal.

Fecha de recepción del abstract: Miércoles, 4 de junio de 2025

Fecha de entrega del informe: Viernes, 6 de junio de 2025

El presente informe validará la traducción del idioma español al inglés si alcanza un porcentaje de: 9 – 10 Excelente.

Si la traducción no está dentro de los parámetros de 9 – 10, el autor deberá realizar las observaciones presentadas en el ABSTRACT, para su posterior presentación y aprobación.

Observaciones:

Después de realizar la revisión del presente abstract, éste presenta una apropiada traducción sobre el tema planteado en el idioma Inglés. Según la rúbrica de evaluación de la traducción en Inglés, ésta alcanza un valor de 9; por lo cual se valida dicho trabajo.

Atentamente



MA. Martha Viveros
Docente responsable del
CIDEN

Anexo 3. Entrevista al ingeniero Hair Lima

Entrevista para una tesis enfocada en el diseño de un sistema de rutas óptimas para la recolección de desechos sólidos

Participantes: 3

Entrevistadores: Cadena Estefanía, Lechón Jorge

Entrevistado: Lima Hair encargado del departamento de ambiente de la ciudad de Tulcán

Introducción:

Es un gusto poder realizar esta entrevista. Mi nombre es Jorge Lechón, estudiante de la carrera de Computación en la Universidad Politécnica Estatal del Carchi, y junto a mi compañera, Estefanía Cadena, estamos llevando a cabo un proyecto de tesis enfocado en el desarrollo de un sistema para generar rutas óptimas en la recolección de desechos sólidos de la ciudad. A través de esta entrevista, buscamos recopilar información clave sobre el funcionamiento del departamento de ambiente y los procesos de recolección, con el fin de comprender mejor los retos y limitaciones actuales en la gestión de desechos sólidos y trabajar hacia un sistema más eficiente y sostenible.

Preguntas:

1. ¿Cuánto tiempo se requiere para completar la recolección total de basura en la ciudad?

En la ciudad de Tulcán, los desechos se recogen a diario en las diferentes áreas urbanas. Sin embargo, en las regiones rurales, este servicio se realiza de manera Inter diaria. La ejecución de esta labor está programada para un total de 7 horas efectivas de trabajo, como se mencionó previamente. Dentro de los límites urbanos, la recolección se realiza todos los días de la semana, desde las 6 a.m. hasta la 1 p.m.

En relación con las parroquias, la recolección de desechos sólidos se lleva a cabo de 1 a 2 veces por semana, siguiendo el mismo horario establecido.

2. ¿Cuál es el tamaño del departamento de ambiente (número de empleados, ingresos anuales, etc.)?

Para llevar a cabo el servicio de recolección de desechos sólidos, es esencial entender el procedimiento. Todo comienza con el personal reuniéndose en un punto de encuentro, donde se congregan tanto los trabajadores como los conductores.

Contamos con 7 carritos recolectores, cada uno con su respectivo chofer, y cada equipo está acompañado por 2 auxiliares que se encargarán de recolectar en una zona específica de la ciudad durante un mes. En el conjunto de vehículos recolectores, se incluyen 5 carritos destinados a la recolección, 1 camión plataforma y 1 camión especializado en el reciclaje.

En total, participan 25 personas que desempeñan roles específicos para garantizar la efectividad y eficiencia del servicio de recolección.

3. ¿Cuál es la estructura organizativa y los departamentos relevantes para el proyecto?

Con los departamentos que se tiene convenios son Jefatura de Tránsito.

4. ¿Cuál es la definición del ámbito de manejo de desechos sólidos?

En la mayoría de las ciudades de Ecuador, el GAD municipal tiene la responsabilidad de garantizar la limpieza de la ciudad, lo que lo convierte en el principal encargado del manejo y tratamiento de los desechos sólidos.

5. ¿Cuáles son las tendencias y desafíos actuales en el sector?

Este departamento ha enfrentado una serie de desafíos, abarcando aspectos desde lo económico hasta lo técnico. En primer lugar, el equipo encargado de proporcionar este servicio presenta una edad considerable, mayoritariamente superando los 50 años. Esta dinámica ha impactado negativamente en la eficiencia del servicio, siendo señalado como parcialmente efectivo.

Otro aspecto relevante es que los ciudadanos abonan una tarifa mínima por el servicio, aunque se considera la posibilidad de implementar una tarifa más elevada con el tiempo para optimizar la calidad del servicio. A esto se suma el desafío de que los recursos, como los camiones se encuentran en la etapa final de su vida útil, lo que los ha dejado desactualizados.

En relación con el pago por el servicio de recolección, previamente el municipio cubría el 60% del subsidio. Sin embargo, este porcentaje ha disminuido al 44%,

resultando en que cada usuario ahora abone 0.28 centavos mensuales, reflejados en la factura de electricidad.

Otro inconveniente significativo es la cobertura del servicio en la ciudad, estimada en un 95%. No obstante, en las zonas rurales, que incluyen ocho parroquias, la cobertura se reduce al 60%, generando insatisfacción entre los usuarios.

6. ¿Cuáles son las normativas y regulaciones relevantes?

La prestación del servicio de recolección de basura representa un costo de 2 millones de dólares. Aunque cuenta con un subsidio del 44%, se proyecta que este gasto pueda reducirse con el tiempo.

Las normativas asociadas a este departamento comprenden:

GAD municipales responsables del tratamiento de desechos sólidos.

La Ley Orgánica de Economía Circular está diseñada para fomentar el reciclaje. Asimismo, existen ordenanzas municipales que regulan aspectos relacionados con los desechos sólidos y establecen normativas en cuanto al cobro por el servicio de recolección

7. ¿Cuál es la descripción del sistema actual para la recolección de desechos sólidos?

"En la ciudad, se ha instaurado un sistema de recolección mediante contenedores, facilitando que los ciudadanos depositen sus desechos sólidos en puntos específicos asignados para este fin.

Por otro lado, existe un método adicional conocido como "A pie y vereda". En áreas donde no hay contenedores de basura, los carros recolectores deben esperar la disponibilidad para realizar la recolección directamente a pie y en las veredas.

8. ¿Cuáles son los métodos empleados para planificar las rutas de recolección?

Se utiliza una metodología fundamentada en una planificación mensual. A modo ilustrativo, consideremos el código 0001, habitualmente asignado a un conductor. A este se le asignan dos personas adicionales, denominadas auxiliares, quienes operarán en una zona específica de la ciudad durante todo un mes. La recolección se lleva a cabo en su totalidad por la mañana para evitar congestiones y reducir el impacto en el tráfico. No obstante, en situaciones excepcionales, como averías de vehículos, se ajusta el horario para realizar la recolección durante las horas nocturnas

9. ¿Cuáles son las tecnologías o sistemas existentes para la gestión de rutas?

La gestión actual carece de un enfoque formal y, en cambio, se sustenta en métodos empíricos. Para recopilar datos sobre la duración y la ruta del servicio, se ha implementado la colocación de un sistema GPS en el recolector, generando así un mapa detallado del recorrido. Esta recopilación de información se lleva a cabo cada dos meses. Asimismo, anualmente se realiza una evaluación del peso de la basura con el objetivo de identificar qué ruta contribuye más significativamente a la contaminación.

10. ¿Cuáles son los datos disponibles sobre los desechos sólidos (volumen, tipo, ubicación, etc.)?

Desde 2019, se han realizado actividades de caracterización de desechos sólidos, pesaje, aplicación de fórmulas para calcular el volumen y monitoreo de la producción de desechos sólidos por habitante. Al principio, la generación era de 0.63 kg por persona al día, pero actualmente ha aumentado a 0.72 kg por persona.

11. ¿Cuáles son los datos históricos de recolección de desechos sólidos (rutas, tiempos, eficiencia, etc.)?

Página web verificar SNIN

12. ¿Qué sistemas de información geográfica (SIG) o software son empleados para la planificación de los recorridos?

No cuentan, pero se ha realizado un seguimiento de las rutas con una app

13. ¿Cuáles son las herramientas de gestión y monitoreo de rutas existentes?

No existe ningún, verificación de ruta por GPS

14. ¿Cuáles son los dispositivos de seguimiento y comunicación utilizados en los vehículos?

No tiene, la comunicación es vía celular

15. ¿Cuáles son las limitaciones técnicas, operativas o financieras actuales?

La mayor limitación que se tiene es con respecto a lo financiero, debido a que los carros recolectores tienden a dañarse con el cual es un gasto considerable.

Tabla 18. Vehículos utilizados en la recolección de desechos sólidos.

Vehículos	Año
2	2014
1	2021
4	Vehículos considerados antiguos

16. ¿Cuáles son las restricciones geográficas o de infraestructura que puedan afectar la planificación de rutas?

En las zonas rurales de la ciudad, la distancia se convierte en una limitación para ofrecer un servicio eficiente. En el peor de los casos, se debe recorrer hasta 80 km, generando costos adicionales. En comparación, en la ciudad, la distancia máxima que se alcanza a recorrer es de 40 km.

17. ¿Cuáles son los factores ambientales relevantes?

Los factores que influyen son los días de invierno debido a que la recolección se complica por dispersión de la basura

18. ¿Cuáles son los principales problemas ambientales asociados a la recolección y manejo de desechos sólidos?

Un desafío considerable se encuentra en la aparición de zonas susceptibles a infecciones, lo que representa un peligro para la salud de los habitantes.

19. ¿Cuáles son las tendencias en la gestión de desechos sólidos?

Leyes propuestas por el país, reciclaje todo esto enfocado en la ley circular

20. ¿Qué estrategias son más efectivas para el manejo adecuado de los desechos sólidos?

Fomentar la concienciación ciudadana para evitar la contaminación y promover prácticas responsables en la comunidad.

21. ¿Cuáles son los factores que se consideran al planificar las rutas de recolección?

Que las personas tengan un lugar bueno para depositar su basura y sea accesible para los carros y personal de recolección.

22. ¿De qué manera se evalúa la efectividad en la recolección de desechos sólidos?

Se realiza el pesaje de los desechos sólidos recolectados para determinar la cantidad exacta, con el propósito de analizar la eficiencia del proceso de recolección e identificar posibles inconvenientes.

23. ¿Cuáles son los principales desafíos en la recopilación de datos sobre desechos sólidos?

La dificultad radica en que el cumplimiento de la ruta, el pesaje y la eficiencia del servicio no se abordan diariamente, sino que simplemente se consideran como completados.

24. ¿Cuáles son las principales limitaciones de las tecnologías y herramientas existentes?

Una limitación está ligada con la falta de presupuesto para adquirir herramientas tecnológicas.

25. ¿Qué innovaciones y avances tecnológicos están marcando tendencia en el desarrollo de herramientas para la gestión de desechos sólidos?

Se ha propuesto tecnología para los carros chip de seguimiento, pero por temas económicos no se ha podido implementar

26. ¿Cuáles son los principales desafíos y limitaciones para implementar un sistema de rutas óptimas?

Se busca una mejora integral del sistema con la aspiración de establecer una empresa de servicios de limpieza. No obstante, la viabilidad de esta idea no es compartida por todos debido a restricciones económicas. A pesar de este desafío, se evidencia un progreso constante en la calidad del servicio, impulsado por los pagos que el municipio realiza, contribuyendo así a una mejora continua.

27. ¿Cómo cree que se pueden superar estos desafíos y limitaciones?

Voluntad política, concientización de la ciudadanía.

Proceso de recolección

El horario de recolección se extiende desde las 6 a.m. hasta la 1 p.m. La reunión inicial se realizará en el garaje del municipio, cerca del Parque Ayora, y finalizará en el relleno sanitario ubicado en la zona suroriental, cerca del colegio militar.

El servicio de recolección se realiza de dos maneras principales: a pie y por veredas, complementado con recolección vehicular. En zonas de difícil acceso, se utiliza una volqueta para garantizar la eficiencia y efectividad del proceso. A continuación se detalla las zonas de recolección con su alcance:

Tabla 19. Tipos de recolección

Zona	Sector, avenidas	Tipo de recolección
1	Rumichaca – Argentina colocar una referencia de localización en las imágenes de rutas que tenemos	A pie y vereda
2	Argentina - Boyacá	Contenerizado
3	Boyacá – Unión 2 calles	Contenerizado
4	Unión 2 calles - Andrés Bello	Contenerizado
5	Andrés Bello - Obelisco	Contenerizado
6	Parroquias: Julio Andrade, Santa Marta, Carmelo y otras	

En la actualidad, operan siete vehículos de recolección, cada uno asignado a un conductor y acompañado por dos auxiliares. Este equipo se dedica a una zona de recolección durante un mes, rotando al inicio de cada nuevo mes. A lo largo del año, abarcan las seis zonas de recolección, garantizando así la cobertura completa en todas ellas.

Anexo 4. Distribución de rutas por zona en la ciudad de Tulcán

RUTA N° 1 DE RECOLECCIÓN TULCÁN

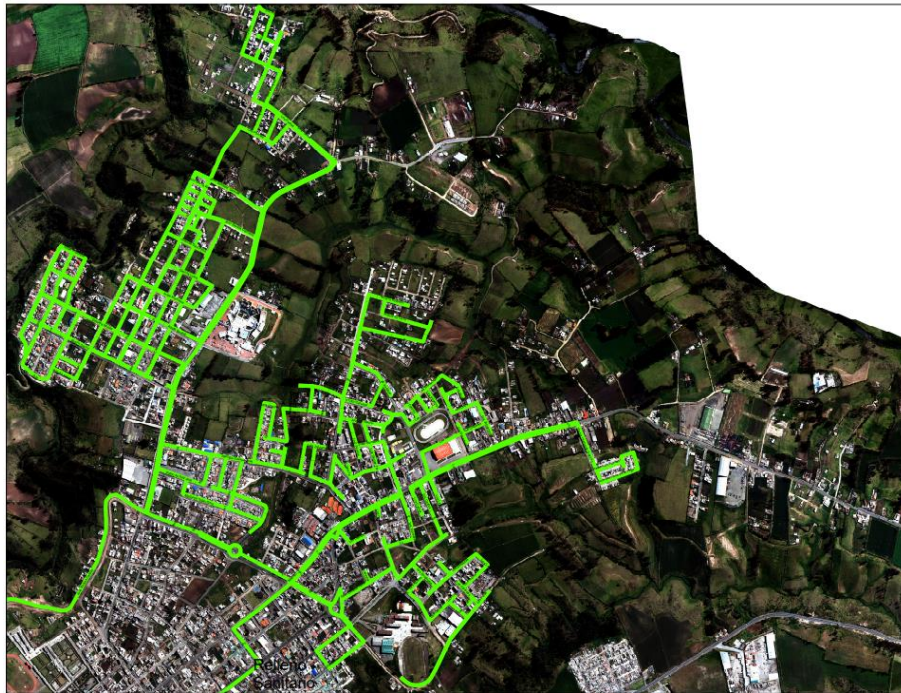


Figura 59. Zona de recolección 1

RUTA N° 2 DE RECOLECCIÓN TULCÁN



Figura 60. Zona de recolección 2

RUTA N° 3 DE RECOLECCIÓN TULCÁN

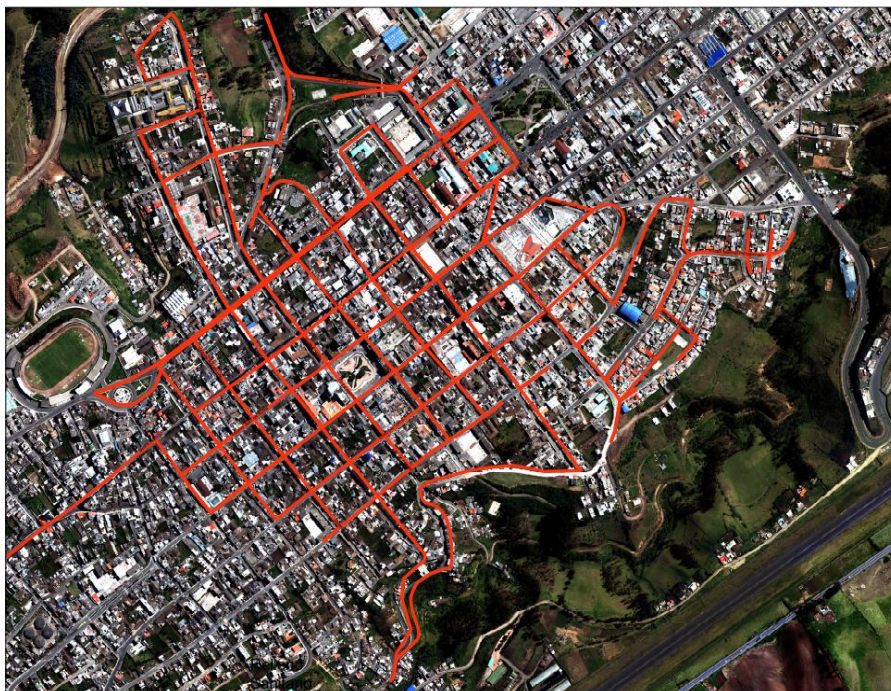


Figura 61. Zona de recolección 3

RUTA N° 4 DE RECOLECCIÓN TULCÁN



Figura 62. Zona de recolección 4

RUTA N° 4 DE RECOLECCIÓN TULCÁN



Figura 63. Zona de recolección 5

Zona Julio Andrade



Figura 64. Zona Julio Andrade

Anexo 5. Cuestionario aplicado en la ciudad de Tulcán.

1. ¿Considera que el sistema de georreferenciación empleado en la recolección de basura es preciso?
 - a) Muy en desacuerdo
 - b) En desacuerdo
 - c) Indeciso
 - d) De acuerdo
 - e) Muy de acuerdo

2. ¿Sabe que es un sistema de georreferenciación?
 - a) Muy en desacuerdo
 - b) En desacuerdo
 - c) Indeciso
 - d) De acuerdo
 - e) Muy de acuerdo

3. ¿Cree que el municipio de la ciudad emplea la georreferenciación para llevar a cabo la recolección de desechos sólidos?
 - a) Muy en desacuerdo
 - b) En desacuerdo
 - c) Indeciso
 - d) De acuerdo
 - e) Muy de acuerdo

4. ¿Consideraría que los días de recolección de la basura en la ciudad son adecuados?
 - a) Muy en desacuerdo
 - b) En desacuerdo
 - c) Indeciso
 - d) De acuerdo
 - e) Muy de acuerdo

5. ¿En el área en la ciudad donde usted reside la recolección de basura es eficaz?
- a) Muy en desacuerdo
 - b) En desacuerdo
 - c) Indeciso
 - d) De acuerdo
 - e) Muy de acuerdo
6. ¿Se encuentra cómodo y satisfecho con la recolección de basura en la ciudad?
- a) Muy en desacuerdo
 - b) En desacuerdo
 - c) Indeciso
 - d) De acuerdo
 - e) Muy de acuerdo
7. ¿Consideraría que la ubicación de los contenedores de basura de la ciudad es adecuada?
- a) Muy en desacuerdo
 - b) En desacuerdo
 - c) Indeciso
 - d) De acuerdo
 - e) Muy de acuerdo
8. ¿Cree usted que la cantidad de contenedores de basura disponibles en la ciudad es suficiente?
- a) Muy en desacuerdo
 - b) En desacuerdo
 - c) Indeciso
 - d) De acuerdo
 - e) Muy de acuerdo
9. ¿Las rutas de recolección de basura cubre todos los barrios de la ciudad?

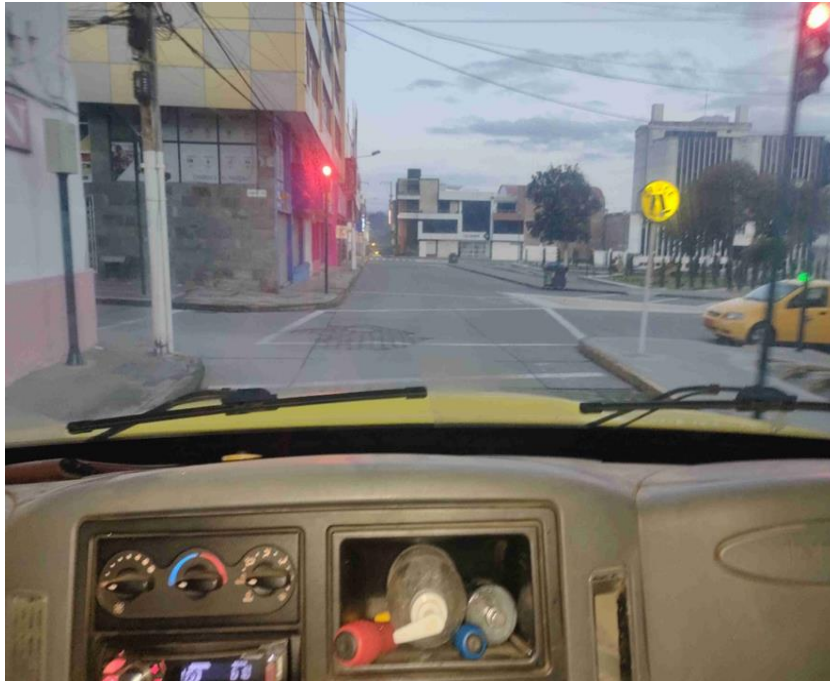
- a) Muy en desacuerdo
- b) En desacuerdo
- c) Indeciso
- d) De acuerdo
- e) Muy de acuerdo

10. ¿Se encuentra satisfecho con la frecuencia que se realiza la recolección de basura en su zona?

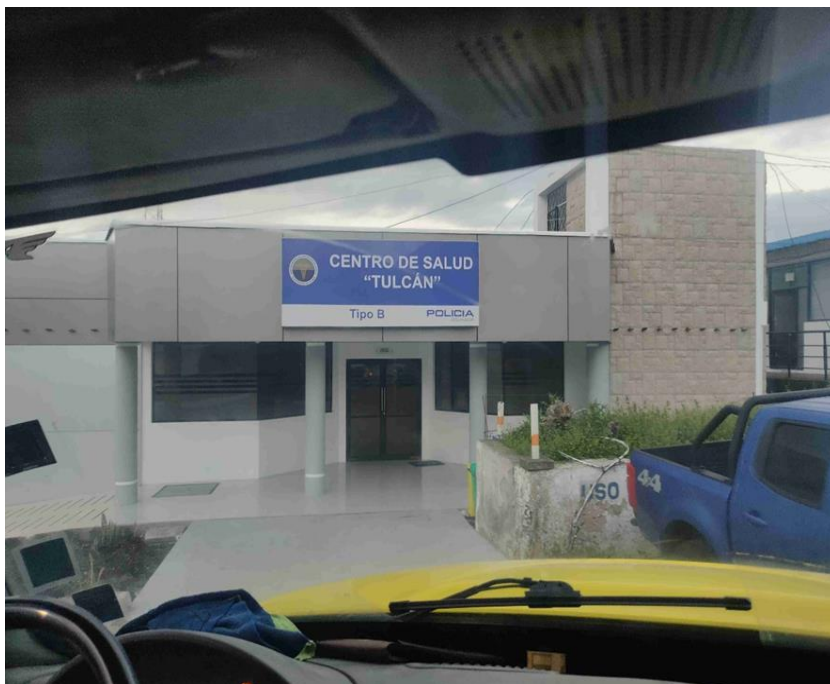
- a) Muy en desacuerdo
- b) En desacuerdo
- c) Indeciso
- d) De acuerdo
- e) Muy de acuerdo

Anexo 6. Evidencias de la recolección Zona 2.

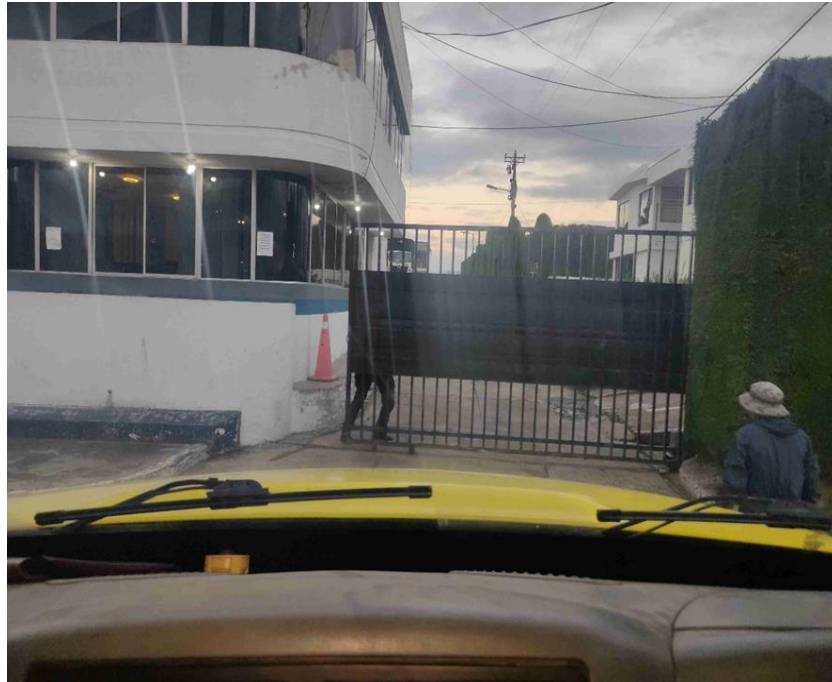
- Evidencia del paso del camión recolector – Zona 2 Intersección entre la calle y cuál es la avenida Diez de agosto y J, J Olmedo – Punto de partida y avenida definir bien eso



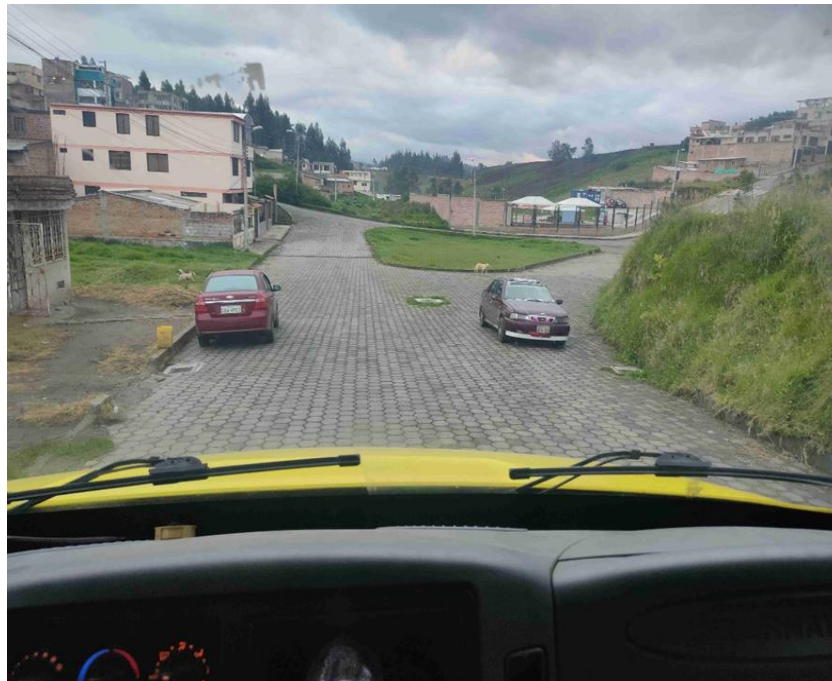
- Evidencia del paso del camión recolector – Zona 2 por el Área de salud al interior de la Sub-Zona de Policía Carchi nº4



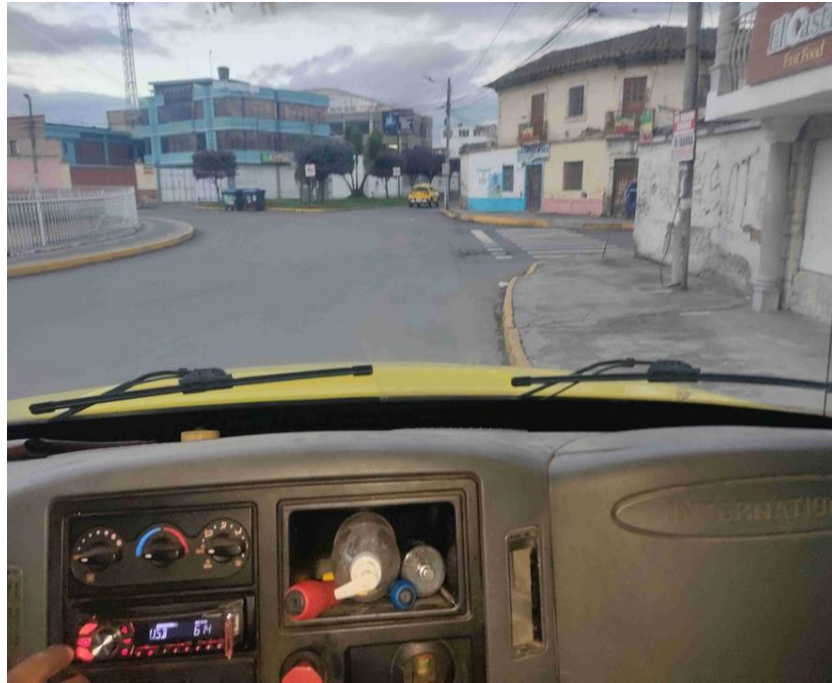
- Evidencia del paso del camión recolector – Zona 2 Sub-Zona de Policía Carchi n°4



- Evidencia del paso del camión recolector – Zona 2 Intersección entre avenida los Ciprés y la avenida los Álamos



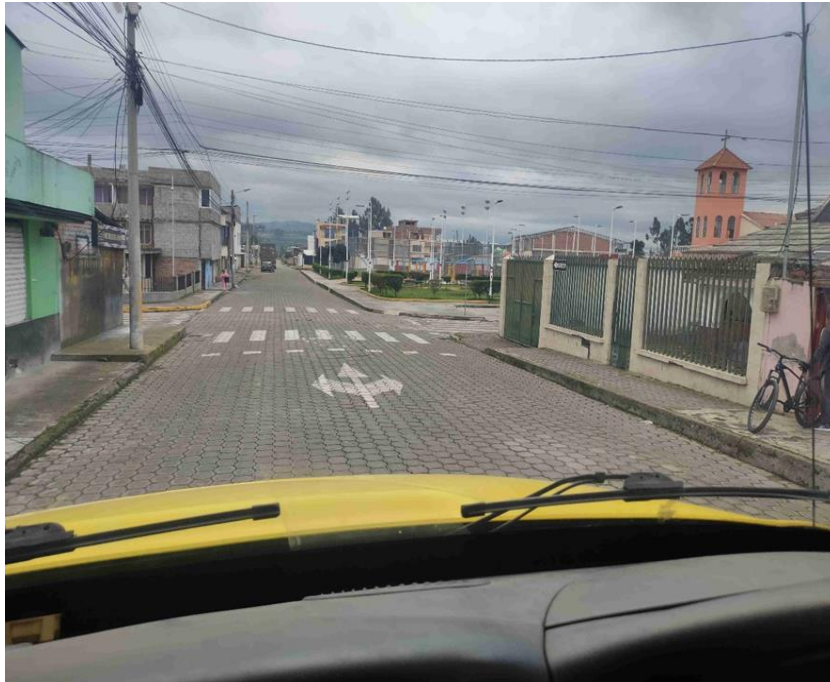
- Evidencia del paso del camión recolector – Zona 2 Intersección entre la avenida Guatemala y avenida Coral



- Evidencia del paso del camión recolector – Zona 2 Avenida San Francisco y Avenida Rubén Darío



- Evidencia del paso del camión recolector – Zona 2 Intersección entre la avenida Adolfo Becker y avenida Miguel de Cervantes



- Evidencia del paso del camión recolector – Zona 2 Intersección en el conjunto 10 de mayo



- Evidencia del paso del camión recolector – Zona 2 Intersección de la avenida Manabí y la Avenida Argentina



Anexo 7. Aplicación del algoritmo de Bellman Ford

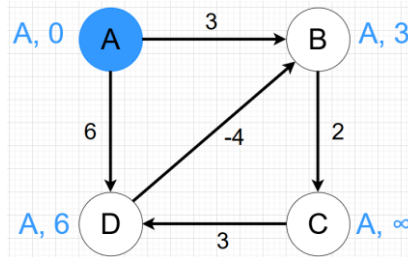


Figura 65. Inicialización de la aplicación del algoritmo de Bellman Ford.

Tabla 20. Actualización de distancias asignadas a cada nodo

Matriz	K=0	K=1	K=2	K=3	K=4	K=...
A	A,0	∞	∞	∞	∞	∞
B	A,3	∞	∞	∞	∞	∞
C	A, ∞	∞	∞	∞	∞	∞
D	A,6	∞	∞	∞	∞	∞

Actualizamos las etiquetas conociendo las nuevas distancias, esta actualización se hace después de haber terminado otra iteración.

Ejemplo: En la primera iteración a los nodos desde A, verificamos la relación que tiene con otros nodos y para eso podemos hacer un salto, es decir llegar solo a los nodos cercanos, pero en la segunda interacción se puede realizar 2 saltos entonces se actualiza la etiqueta si el valor es menor al que estaba propuesto.

- **iteración 1**

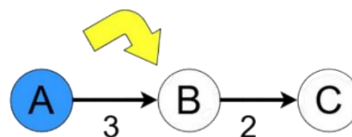


Figura 66. Iteración de A hacia C

En la primera iteración de A hacia C la etiqueta sería (A, ∞).

- **Iteración 2**

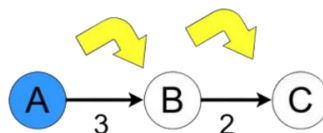


Figura 67. Iteración 2 usando el algoritmo Bellman Ford

En la iteración 2 se puede hacer dos saltos entonces ya tendría relación el nodo A con el nodo C, la etiqueta sería (A, 5).

Siguiendo el grafo propuesto.

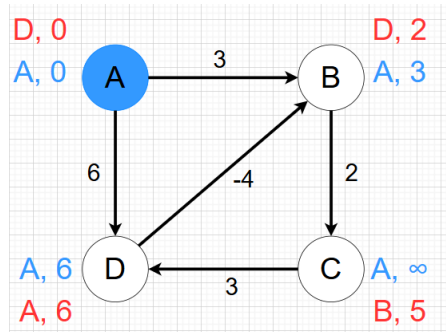


Figura 68. Nueva asignación de distancias

Tabla 21. Actualización de etiquetas según la nueva iteración

Matriz	K=0	K=1	K=2	K=3	K=4	K=...
A	A,0	A,0	∞	∞	∞	∞
B	A,3	D,2	∞	∞	∞	∞
C	A, ∞	B,5	∞	∞	∞	∞
D	A,6	A,6	∞	∞	∞	∞

Actualizamos nuevamente las etiquetas, nueva iteración.

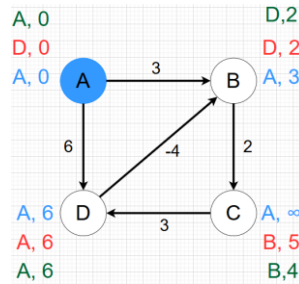


Figura 69. Actualización de las etiquetas en la cuarta iteración

Tabla 22. Cambio de etiquetas

Matriz	K=0	K=1	K=2	K=3	K=4	K=...
A	A,0	A,0	A,0	∞	∞	∞
B	A,3	D,2	D,2	∞	∞	∞
C	A, ∞	B,5	B,4	∞	∞	∞
D	A,6	A,6	A,6	∞	∞	∞

Actualizamos las etiquetas, una vez que verificamos que las etiquetas se repiten el algoritmo finalizará.

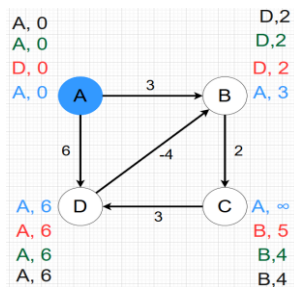


Figura 70. Grafo final

Tabla 23. Etiquetas finales

Matriz	K=0	K=1	K=2	K=3	K=4	K=...
A	A,0	A,0	A,0	A,0	∞	∞
B	A,3	D,2	D,2	D,2	∞	∞
C	A, ∞	B,5	B,4	B,4	∞	∞
D	A,6	A,6	A,6	A,6	∞	∞

Anexo 8. Aplicación del algoritmo Floyd Warshall

Tabla 24. Inicialización del algoritmo de Floyd Warshall

Matriz	0	1	2	3
0	0	3	∞	7
1	8	0	2	∞
2	5	∞	0	1
3	2	∞	∞	0

Iteraciones principales

El algoritmo utiliza tres bucles anidados. En cada iteración, verifica si agregar un nodo intermedio k mejora la distancia entre dos nodos i y j.

Para cada nodo intermedio k:

- Para cada par de nodos i y j:

Actualiza $dist[i][j]$ como:

$$dist[i][j] = \min (dist[i][j], dist[i][k] + dist[k][j])$$

Ejemplo:

Para resolver un grafo con este algoritmo debemos hacerlo con una matriz distancia y una matriz de recorrido.

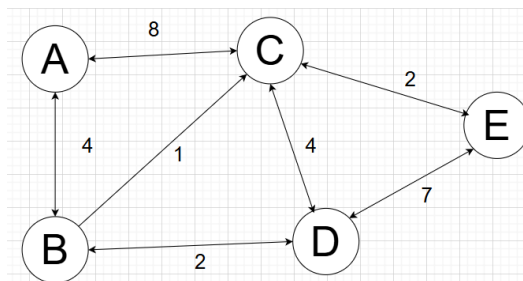


Figura 71. Grafo a aplicar el algoritmo de Floyd Warshall.

Resolución del grafo mediante el modelo de Floyd Warshall: Entonces la solución para el grafo se debe verificar las relaciones que se tiene entre nodos y colocar sus pesos en la matriz distancia, los nodos que no tengan relación se coloca como indeterminado.

Tabla 25. Matriz de distancia algoritmo de Floyd Warshall.

		Matriz distancia				
		A	B	C	D	E
A		0	4	8	∞	∞
B		4	0	∞	2	∞
C		8	1	0	4	2
D		∞	2	4	0	7
E		∞	∞	2	7	0

Para cada nodo intermedio k, verificamos si pasar por k reduce la distancia entre los nodos i y j. Actualizamos la matriz de distancias usando:

$$dist[i][j] = \min(dist[i][j], dist[i][k] + dist[k][j])$$

Tabla 26. Distancia entre i y j de - Floyd Warshall.

		Matriz recorrida				
		A	B	C	D	E
A		∞	B	B	B	C
B		A	∞	C	D	C
C		A	D	∞	D	E
D		B	B	B	∞	C
E		C	D	C	C	∞

Anexo 9. Aplicación del algoritmo de A*

Dado un grafo con los nodos A, B, C, D, E, F, debemos encontrar el camino óptimo desde A hasta F usando el algoritmo **A***.

Tabla 27. Conexiones existentes - Algoritmo de A*

Nodo	Conexión
A	B (1), C (3)
B	D (3), E (5)
C	D (1)
D	F (2)
E	F (1)

Tabla 28. Nodos y heurística - Algoritmo de A* compilados en una tabla.

Nodo	Heurística
A	2
B	4
C	3
D	2
E	1
F	0

Inicialización

Abierta: lista con el nodo inicial (A) junto con su $g + h$ (en este caso, $g = 0 + h(A) = 6$).

Cerrada: Lista vacía

Abierta [A (6)]

Cerrada: []

Explorar el nodo con el menor f

Sacamos A (6) de abierta y lo agregamos a cerrada.

Expandimos A: B(1), C(3).

Calculamos $f = g + h$ para cada nodo sucesor:

$$B: g(B) = 1, h(B) = 4 \rightarrow f(B) = 1 + 4 = 5$$

$$C: g(C) = 3, h(C) = 3 \rightarrow f(C) = 3 + 3 = 6$$

Abierta: [B(5), C(6)]

Cerrada: [A]

Expandir B(5)

Sacamos B(5) y lo agregamos a cerrada.

Expandimos B: $D(3), E(5)$.

Calculamos f para cada sucesor:

$$(D: g(D) = 1 (B \rightarrow D), h(D) = 2 \rightarrow f(D) = 4 (3 \rightarrow 2) \dots$$

Expandir B (5) (continuación):

Expandimos B: conexiones a D (3) y E (5).

Calculamos $f = g + h$ para cada sucesor:

Para D:

$$g(D) = g(B) + \text{costo}(B \rightarrow D) = 1 + 3 = 4, h(D) = 2 \rightarrow f(D) = g(D) + h(D) = 4 + 2 = 6$$

Para E:

$$g(E) = g(B) + \text{costo}(B \rightarrow E) = 1 + 5 = 6, h(E) = 1 \rightarrow f(E) = g(E) + h(E) = 6 + 1 = 7$$

Expandir C (6):

Sacamos C (6) de abierta y lo agregamos a cerrada.

Expandimos C: conexión a D (1).

Calculamos f para D:

$$g(D) = g(C) + \text{costo}(C \rightarrow D) = 3 + 1 = 4, h(D) = 2 \rightarrow f(D) = g(D) + h(D) = 4 + 2 = 6$$

El nuevo $g(D) = 4$ no mejora el $g(D)$ ya calculado, por lo que no actualizamos D.

Abierta: [D (6), E (7)]

Cerrada: [A, B, C]

Expandir D (6):

Sacamos D(6) de abierta y lo agregamos a cerrada.

Expandimos D: conexión a F(2).

$$\begin{aligned} \text{Calculamos } f \text{ para } F: g(F) = g(D) + \text{costo}(D \rightarrow F) = 4 + 2 = 6, h(F) = 0 \rightarrow f(F) = g(F) + h(F) \\ = 6 + 0 = 6. \end{aligned}$$

Abierta: [E (7), F (6)]

Cerrada: [A, B, C, D]

Expandir F (6):

- Sacamos F (6) de abierta y lo agregamos a cerrada.
- Como F es el nodo objetivo, hemos encontrado el camino más corto.

Resultado

El camino más corto desde A hasta F es:

$$A \rightarrow B \rightarrow D \rightarrow F$$

Con un costo total de 6.

Anexo 10. Certificación de satisfacción



Alcaldía de Tulcán
Administración 2023 - 2027

CERTIFICACIÓN

JEFATURA DE RESIDUOS SÓLIDOS, certificamos que:

SISTEMA DE GESTIÓN DE RUTAS PARA LA RECOLECCIÓN DE DESECHOS SÓLIDOS.

Como estudiantes de la carrera de computación de la Universidad Politécnica Estatal del Carchi, han demostrado un compromiso excepcional con su formación académica y profesional, desarrollando con éxito el software "En el Entorno visual Studio Code" correspondiente a la distribución de rutas para el área de residuos sólidos, este proyecto no solo refleja su dedicación sino también su capacidad para aplicar sus conocimientos adquiridos a lo largo de su formación en esta casa universitaria.

La Jefatura de Residuos Sólidos promueve el bienestar de todos sus miembros y afronta la necesidad de gestionar sistemáticamente y telemáticamente el servicio de manera eficiente y transparente; Con fin de optimizar este proceso se identificó la oportunidad de desarrollar mediante la App Rutas, facilitando la gestión de Residuos Sólidos del GAD Municipal de Tulcán de manera optimizada todos los procesos que conlleva nuestra dependencia.

En este sentido los tesisistas **Jorge Rubén Lechón Guatemal con CI. 1728049139** y **Lizbeth Estefanía Cadena Castro con CI. 0401992268**, asumieron el desafío de crear una opción que respondiera a estas necesidades, realizando la verificación in situ de las actuales rutas de recolección que nuestro servicio ofrece en nuestra jurisdicción.

Funcionalidades:

La aplicación diseñada para la planificación de rutas destinadas para la recolección de residuos sólidos. Inicia con el ingreso, el usuario el cual debe registrarse en la plataforma, permitiendo asignar roles de acceso, como usuario o administrador.

Dentro del sistema, el rol de administrador tiene la capacidad de gestionar usuarios, vehículos, ciudades y horarios, permitiendo tanto la creación como la edición de estos elementos.

La funcionalidad más destacada de la aplicación es la planificación de rutas, donde se visualizan las distintas zonas de recolección segmentadas., cada zona puede gestionarse de forma individual mediante acciones como establecer puntos de inicio y fin, modificar puntos clave, cambiar prioridades para optimizar la ruta, y añadir información relevante como el vehículo asignado, el horario de recolección, los operadores responsables y el tiempo estimado de servicio.

Toda esta información se presenta en la aplicación con su respectiva representación gráfica de la ruta, y también se puede generar un documento para su impresión.

Este software ha sido diseñado tomando en cuenta las normativas internas de la actual ordenanza vigente de la GIRS y presente del GADMT, mediante el uso de plataforma que facilitará actividades cotidianas del servicio de recolección.

Es todo en cuanto podemos certificar en honor a la verdad facultando a los tesisistas el uso del presente para los fines que considere pertinente.



Formulario de identificación personal
LUIS HAIR LIMA PANTOJA

MSc. Luis Hair Lima Pantoja
JEFE DE RESIDUOS SÓLIDOS GADMT

El Reto es por ti Tulcán!

(593) 062980 400
municipiotulcan@gmtulcangob.ec
www.gmtulcangob.ec
10 de Agosto y Olmedo
Barrio Centro Tulcán - Ecuador

Anexo 11. Manual de usuario

UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI
FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES



CARRERA DE COMPUTACIÓN

Manual de Usuario

GAD Municipal de Tulcán

Instituto de Ambientes

Versión: 1.0

Fecha: mayo 2024

Dentro de la página web encontramos el registro e inicio de sesión, al tratarse de usuarios nuevos se debe registrar para poder acceder. Si el usuario ya se registró solo debe poner sus credenciales en el formulario; correo y contraseña.

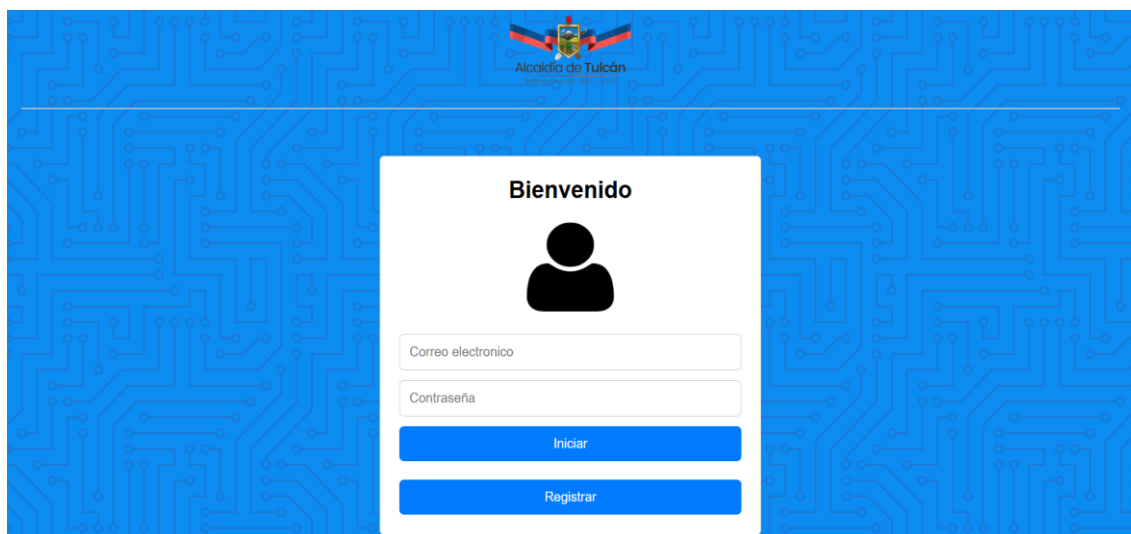


Figura 72. Pantalla de inicio sesión

Registro

El usuario que se va a registrar debe llenar los campos solicitados, en nombre es opcional colocar los 2 nombre, se puede usar solo el primer nombre. Para apellido funciona de la misma forma se pueden colocar los 2 apellidos o solo el apellido paterno, Como siguiente campo tenemos la cedula en donde debemos colocar nuestro número de identificación. Los siguientes campos para llenar son edad, seleccionar la ciudad de donde residimos y colocar nuestro número de celular. Los últimos campos para llenar son primordiales se trata de el correo y contraseña, para el caso del correo va a ser nuestro usuario para acceder al aplicativo acompañado de la contraseña que debe superar los 8 caracteres donde se puede usar una combinación de letras, símbolos o números. El ultimo campo de selección se trata de definir el rol a desempeñar para el usuario donde consta de Administrador, Usuario, Conductor, Auxiliar.

Tabla 29. Roles del aplicativo

Rol	Descripción
Administrador	Tiene acceso completo al aplicativo, puede agregar, eliminar, modificar información de usuarios, vehículos, horarios, ciudad y roles cargados dentro de la base de datos del aplicativo, además de poder planificar, crear rutas.
Usuario	Tiene acceso a visualizar datos cargados en el aplicativo y a planificar, crear rutas.
Conductor	Tiene acceso a visualizar datos cargados en el aplicativo y a planificar, crear rutas.
Auxiliar	Tiene acceso a visualizar datos cargados en el aplicativo y a planificar, crear rutas.

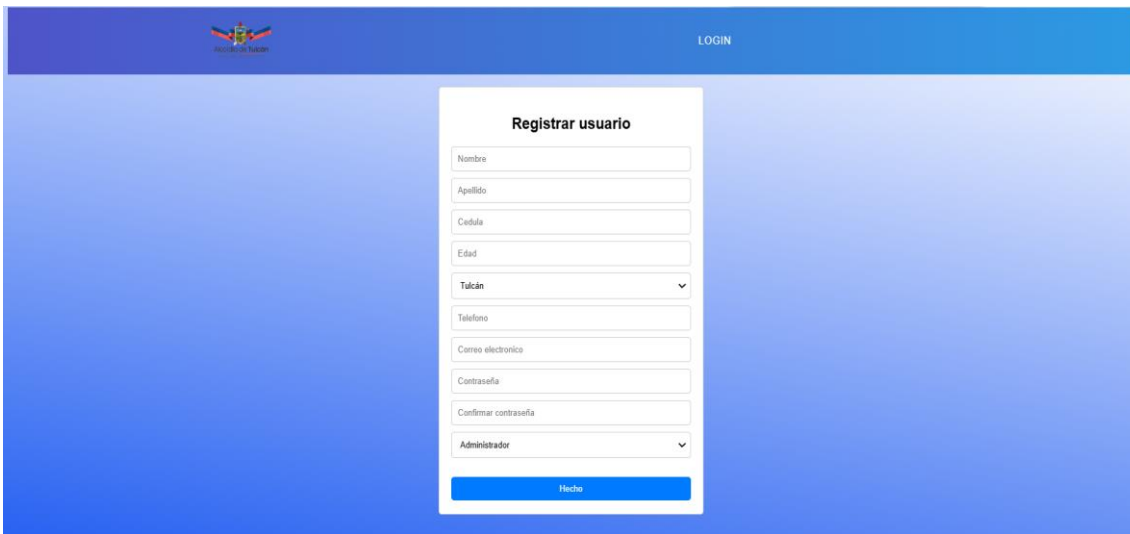


Figura 73. Pantalla de registro

Después de iniciar sesión encontramos el menú principal, donde hay cuatro módulos: Agregar, Información, Rutas y Otros.



Figura 74. Pantalla principal menú

Menú de navegación Servicios que cuenta con atajos a diferentes pantallas del aplicativo.

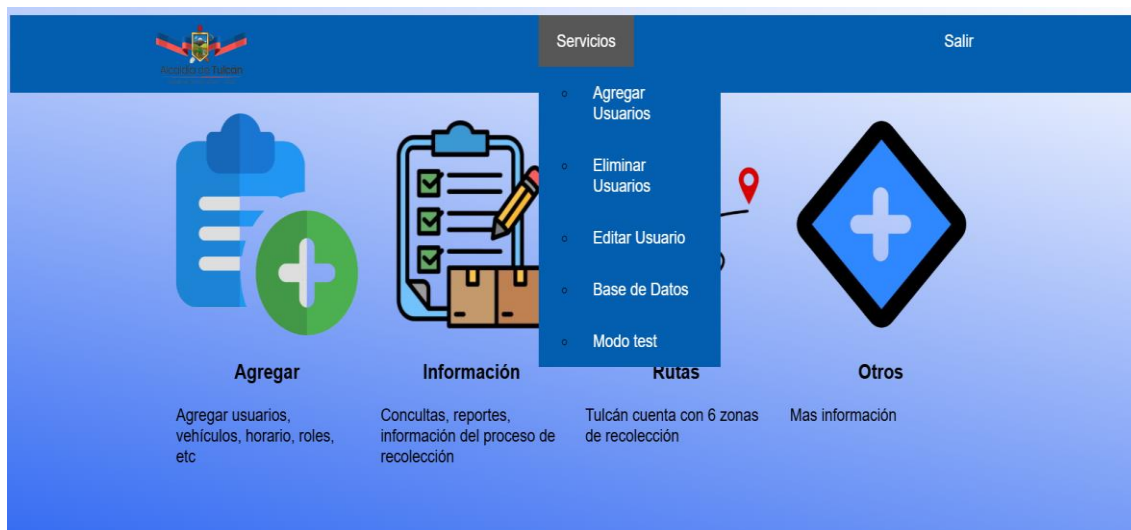


Figura 75. Menú de navegación

AGREGAR

Dentro del primer módulo esta agregar, y las opciones a seleccionar son; agregar, eliminar y editar. La información que se va a manipular es de usuarios registrados, vehículos operativos, horarios establecidos, roles designados y ciudades. Esta información esta almacenada en la base de datos. Seleccionamos Agregar

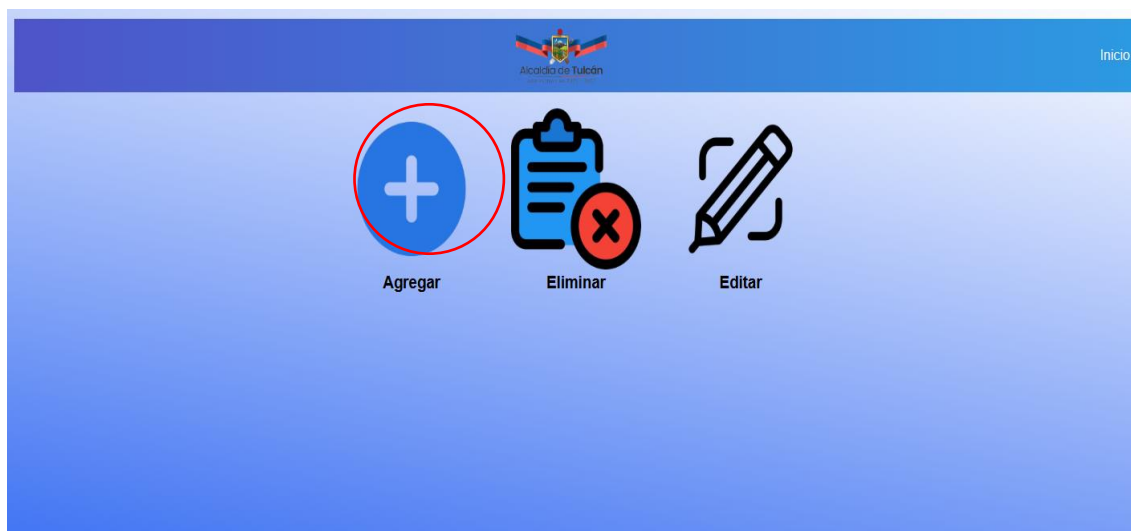


Figura 76. Pantalla agregar

Modulo agregar

Podemos agregar información con su respectivo campo; Usuarios, vehículo, horario, ciudad y roles. Seleccionamos Usuarios.



Figura 77. Pantalla Agregar

Agregar – Usuario

Tenemos la pantalla agregar usuario que es igual a la pantalla registrar usuario, dentro de esta pantalla también podemos registrar nuevos usuarios para que puedan acceder al aplicativo web, entonces se debe llenar los campos solicitados.

Tabla 30. Datos para agregar un nuevo usuario

Cedula	Numero de identidad personal CI
Nombre/Apellido	Podemos agregar nuestro nombre completo
Edad	Colocamos nuestra edad
Ciudad	Seleccionamos la ciudad que residimos
Teléfono	Colocamos nuestro número de celular
Correo	Usamos un correo que sirve como usuario para acceder al aplicativo web
Contraseña	Usamos una contraseña donde podemos combinar números, letras y símbolos.
Rol	Podemos establecer roles para los usuarios, debemos seleccionar entre; administrador, usuario, chofer y auxiliar.

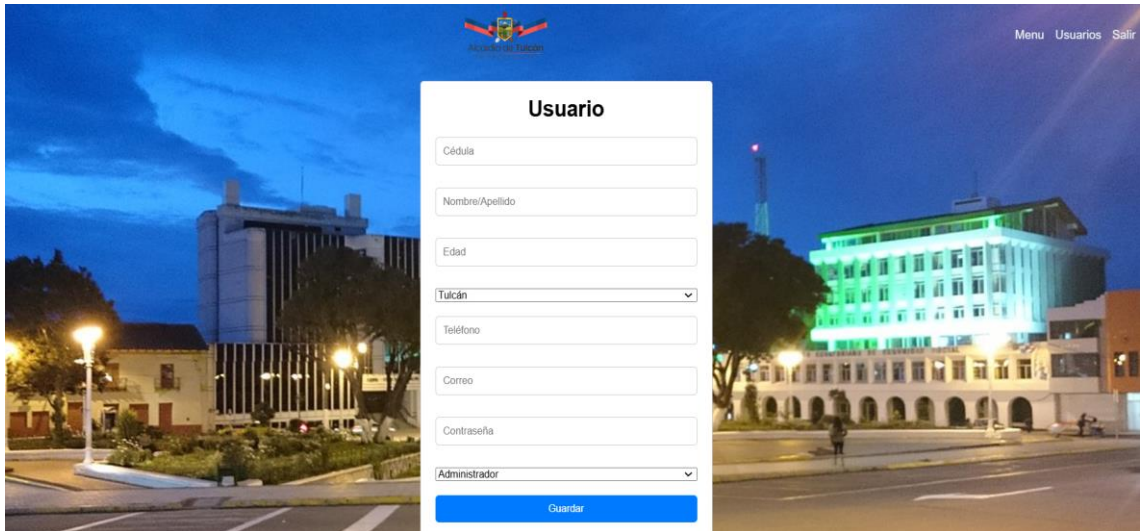


Figura 78. Pantalla agregar usuario

Menú navegación

Tenemos las opciones; Menú, Usuarios, Salir. Estas opciones nos llevan a otras pantallas del aplicativo web.

Menú: Pantalla principal del aplicativo

Usuarios: Una ventana con más opciones para el usuario.

Salir: Cierra sesión del aplicativo web.

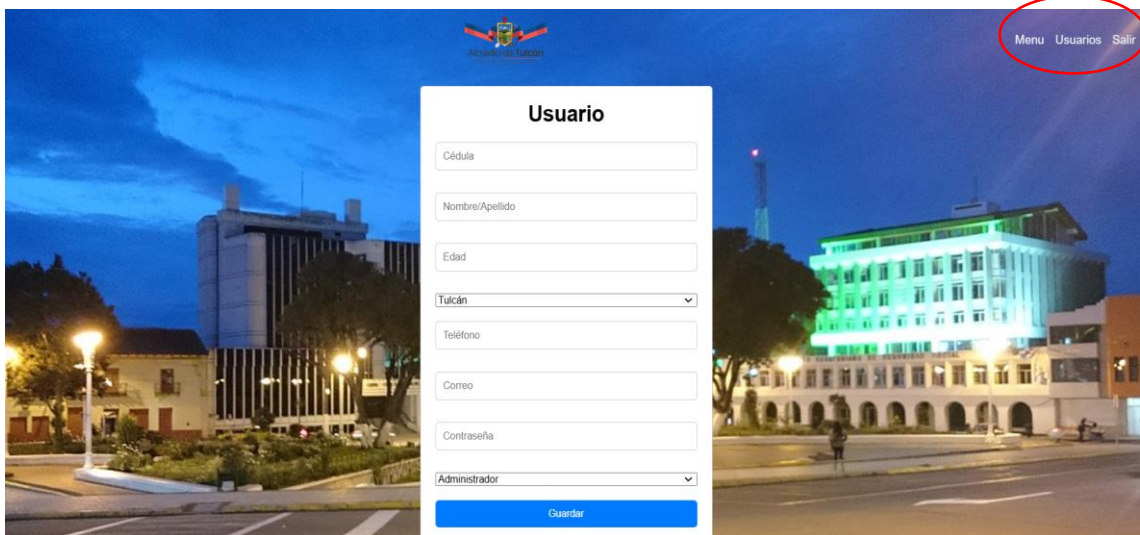


Figura 79. Pantalla agregar usuario datos necesarios

Agregar – Vehículo

Pantalla de Agregar, con las diferentes opciones. Seleccionamos vehículo



Figura 80. Pantalla agregar vehículo

Pantalla de formulario para un nuevo vehículo, ingresamos la información en los campos solicitados, información correspondiente al vehículo.

Tabla 31. Datos asociados a los vehículos de recolección

Tipo de vehículo	Establecemos un tipo que puede ser, administrador o recolector depende la descripción del vehículo.
Marca de vehículo	Ponemos la marca del vehículo.
Código	En las entidades manejan un código para poder identificarlos, colocamos ese código de inventario.
Color	Seleccionamos el color correspondiente del vehículo.
Año	Año de fabricación del vehículo.
Placa	Identificador con la ANT
Carga	Donde está ubicado el contenedor de basura en este vehículo
Estado	Si el vehículo esta activo o descompuesto.

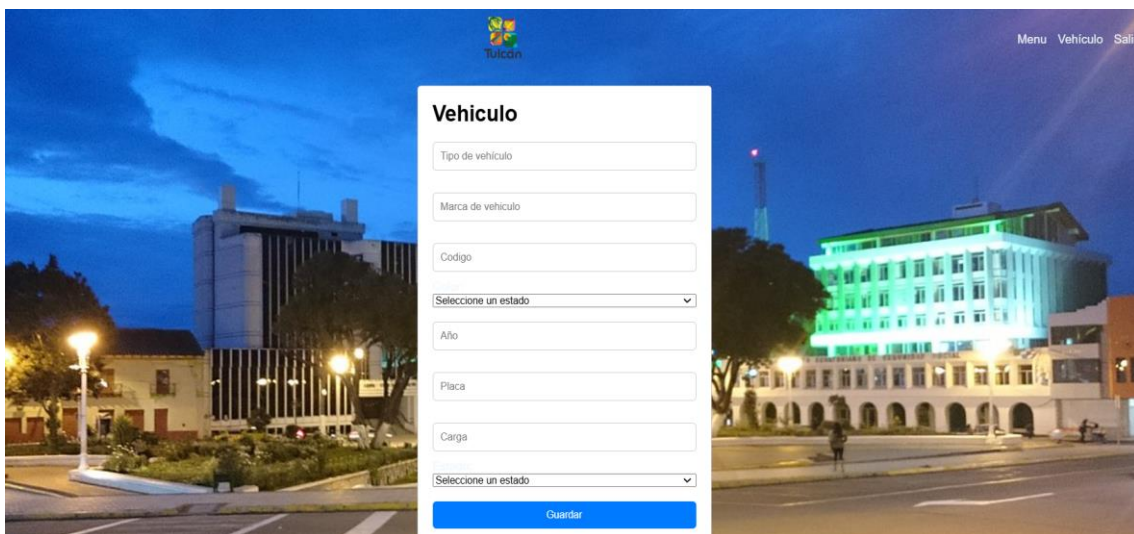


Figura 81. Pantalla agregar vehículo

AGREGAR – HORARIO

Seleccionamos horario para establecer uno nuevo, entonces seleccionamos el icono de reloj.



Figura 82. Pantalla agregar horario

Pantalla de agregar nuevo horario, debemos llenar los campos requeridos; Nombre, Horas. Para nombre colocamos las jornadas laborales y en horas el tiempo de duración de cada jornada.

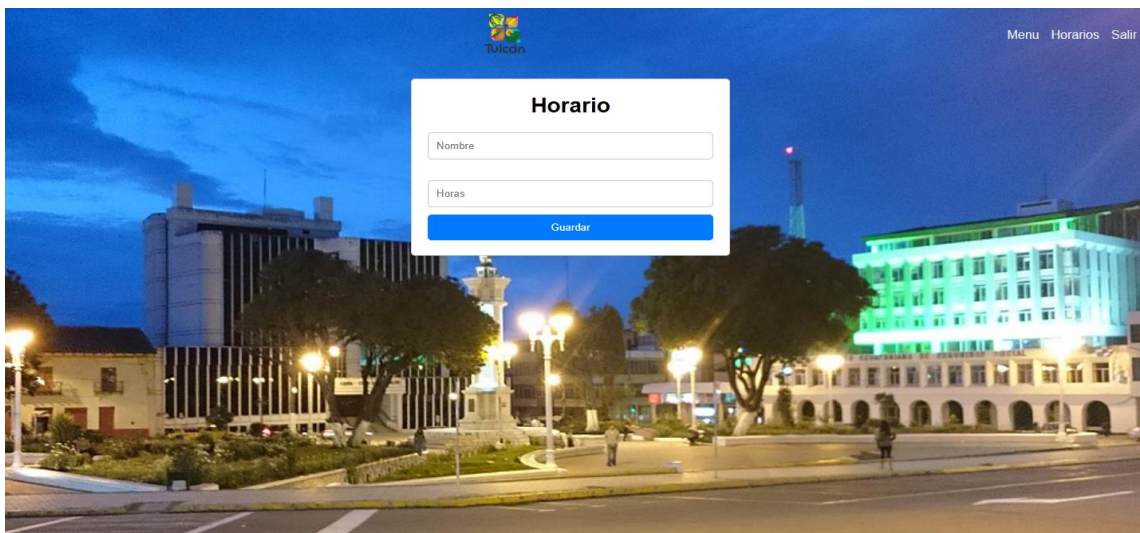


Figura 83. Pantalla agregar horario opción horario

AGREGAR – CIUDAD

Seleccionamos el icono con una ciudad para ingresar una nueva.

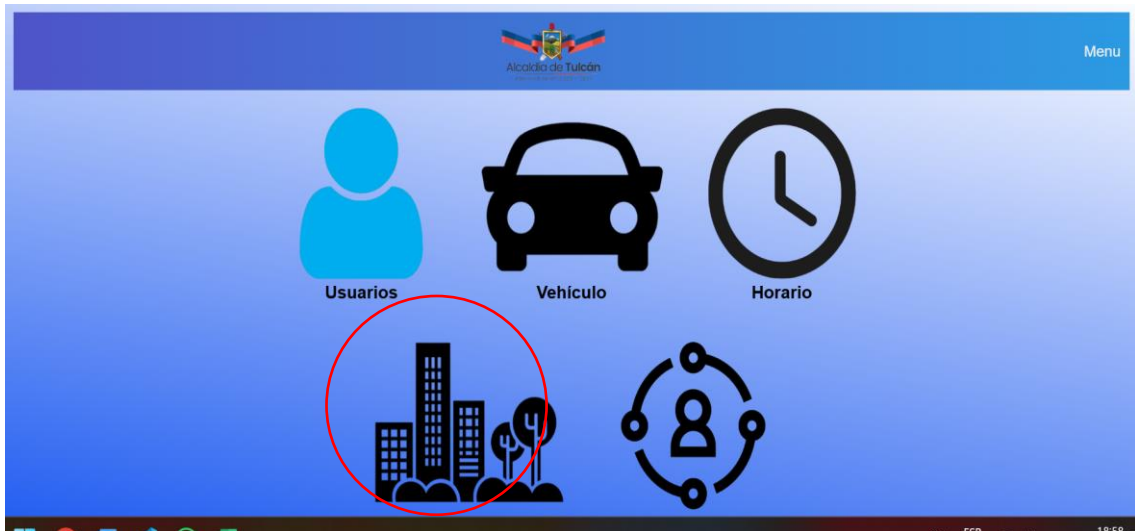


Figura 84. Pantalla agregar ciudad

Ingresamos el campo solicitado ciudad, podemos ingresar una ciudad que no esté registrado en la base de datos.

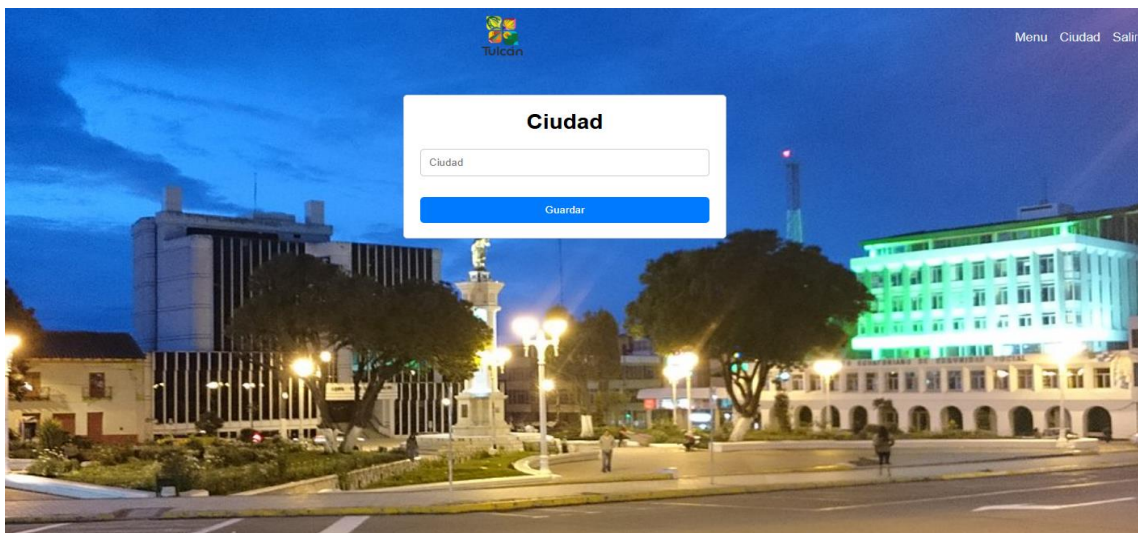


Figura 85. Pantalla agregar ciudad

AGREGAR – ROL

Seleccionamos el icono rol para agregar uno nuevo.

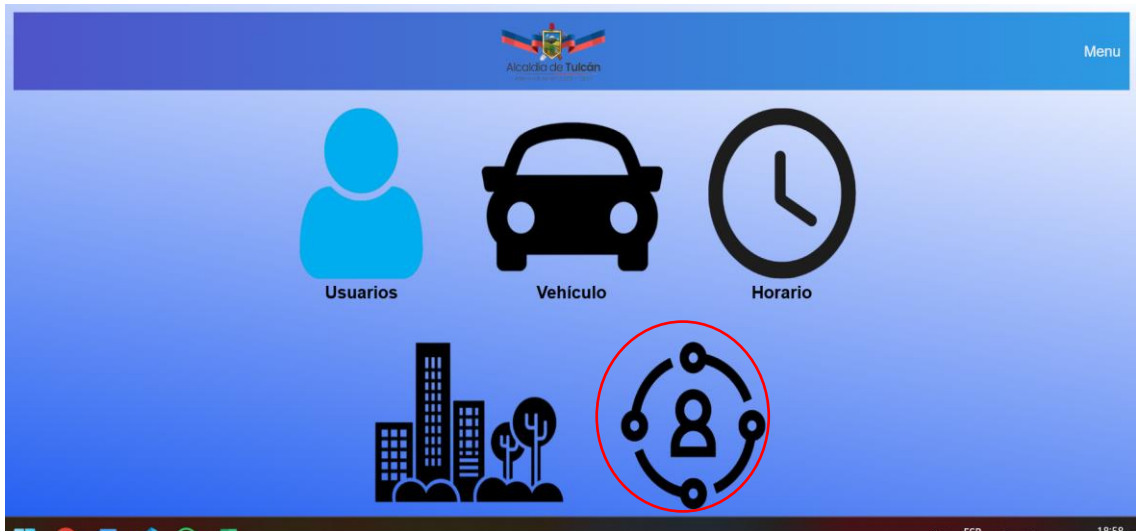


Figura 86. Pantalla agregar rol

Ingresamos información en el campo solicitado, el aplicativo tiene definido cuatro roles; administrador, usuario, chofer y auxiliar.

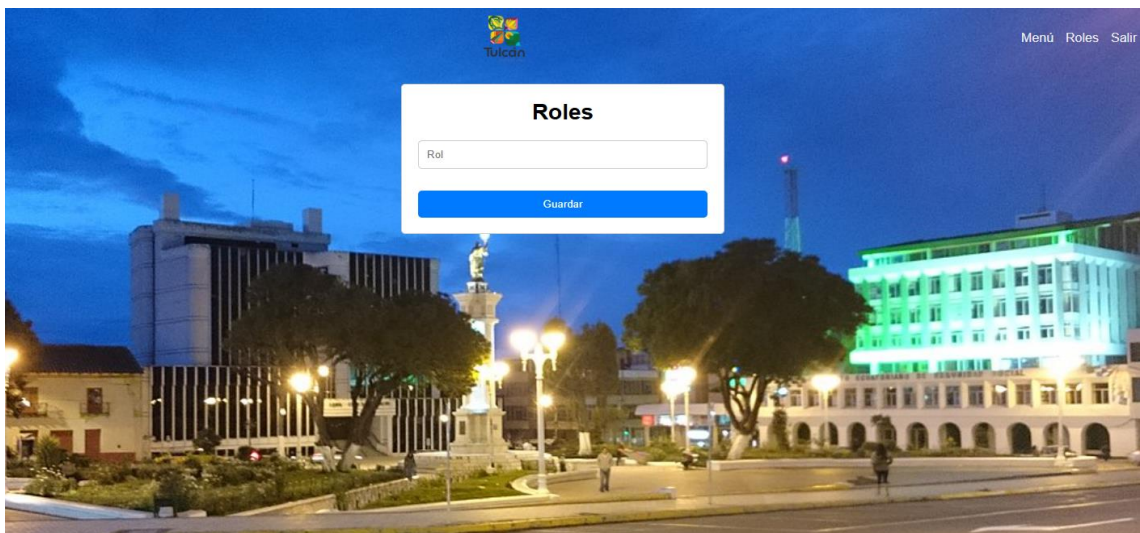


Figura 87. Pantalla agregar rol

ELIMINAR

Opción eliminar, proporciona acceso a los datos como; usuario, vehículo, horario, ciudad y rol donde podemos eliminar información.

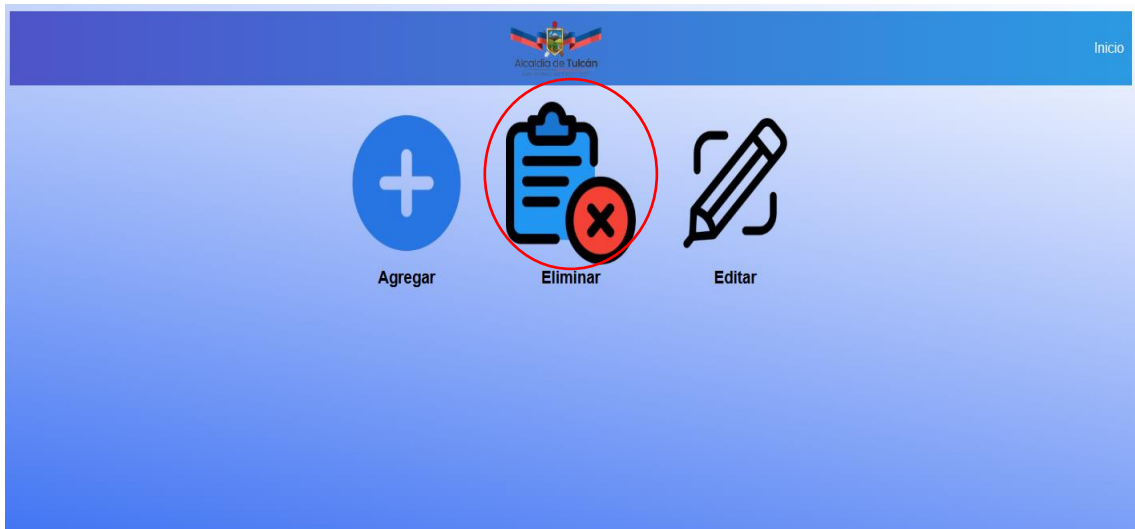


Figura 88. Pantalla modulo eliminar

Dentro del módulo eliminar tenemos otro menú con las diferentes opciones similares al de agregar, pero en esta sesión las opciones proporcionadas serán eliminar información.

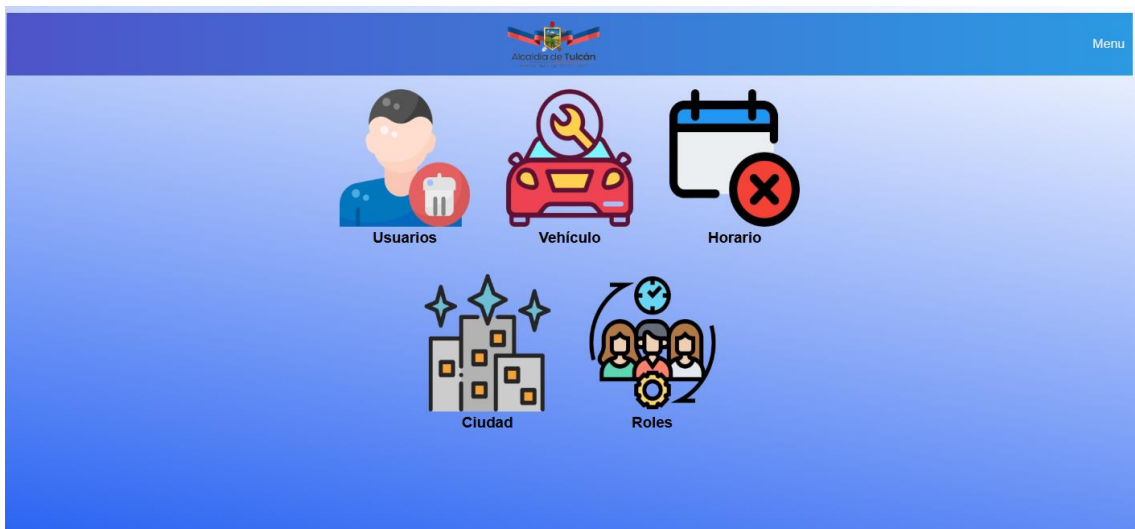


Figura 89. Pantalla eliminar

Seleccionamos el icono de usuario para acceder a su ventana y eliminar algún usuario que no debe constar en la base de datos.

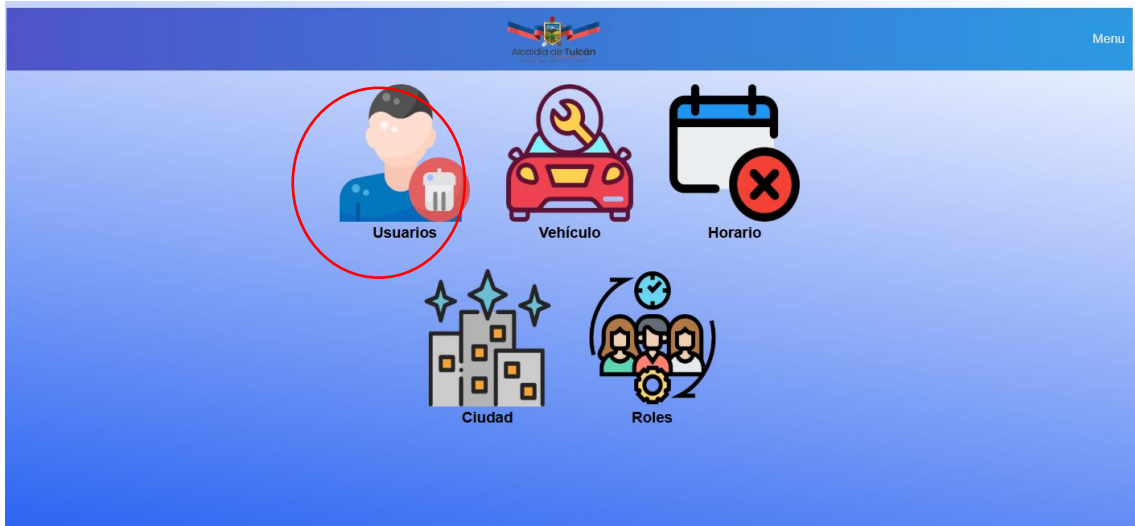


Figura 90. Pantalla eliminar usuario

En la ventana eliminar usuario tenemos la información de todo el registro de los usuarios en una tabla, se visualiza datos como; cedula, nombre, edad, ciudad, teléfono, correo.

Se puede eliminar un usuario al digitar en el número de cédula en el panel "Eliminar usuario" una vez cumplido esta función saldrá un mensaje de operación realizada y se actualizará la tabla usuarios.

Cedula	Nombre	Edad	Ciudad	Teléfono	Correo	Contraseña
0987654321	Estefania Cadena	22	Tulcán	0979405445	cadenacastro@gmail.com	\$2a\$13\$JWx84nZ3kPUeCqh.B/CK0u9/X8o8p/H7g4ugOPGSXZbx8ALmKyTPa
1234567888	Wilmer Colimba	23	Tulcán	0987654322	wilmer@gmail.com	\$2a\$13\$k2K6zj5KmpdyfpGbdOSsB.GwL.DaPZUWrJkWs1xRhE.vs.nT60Nm
1298645371	Estefania Tipaz	22	Tulcán	0974635182	estefania@gmail.com	\$2a\$13\$CJGw3otnhtCjarWBgnByaessoPyLBp1edA7oW0ojoL.0TQviGVC
1723456789	Alejandro Cifuentes	27	Tulcán	0912345678	www.alejandrocifuentes@gmail.com	\$2a\$13\$MqzPWF23eimuoUfBor1VTOHoKbum6vGOoJ2nBOVC5R9doKUip.N8.
1728049139	Jorge Lechon	23	Tulcán	0979405444	jorgeguatemala17@gmail.com	\$2a\$13\$5cG9z.sTCikdvV406/cDfeV5kR.tN0KbO2uRC0mDK2nsZgxy6UEe

Figura 91. Pantalla eliminar usuario

Eliminar – vehículo

Seleccionamos el icono de vehículo para acceder a su ventana.



Figura 92. Pantalla eliminar vehículo

Dentro de la pantalla eliminar vehículo notamos que la función eliminar es similar a eliminar usuario, tenemos la información correspondiente a vehículo; Id, tipo, marca, código, color, año, carga y estado.

Colocamos el id del vehículo que debemos eliminar del registro, una vez terminada la operación se notificara con un mensaje que la operación está realizada y se actualizara la tabla de vehículo.

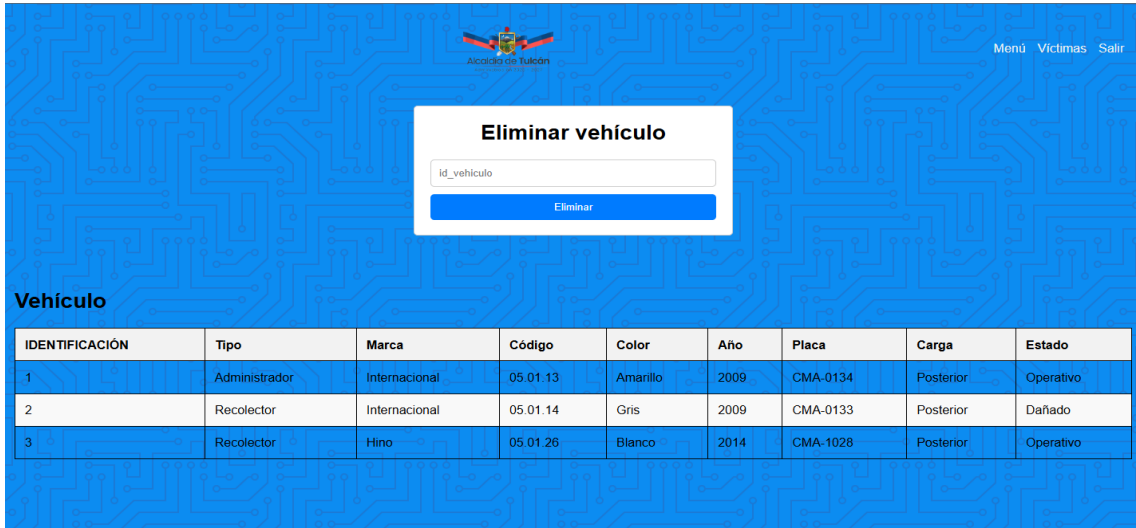


Figura 93. Pantalla eliminar vehículo con id

Eliminar – Horario

Seleccionamos el icono de horario para acceder a su ventana.

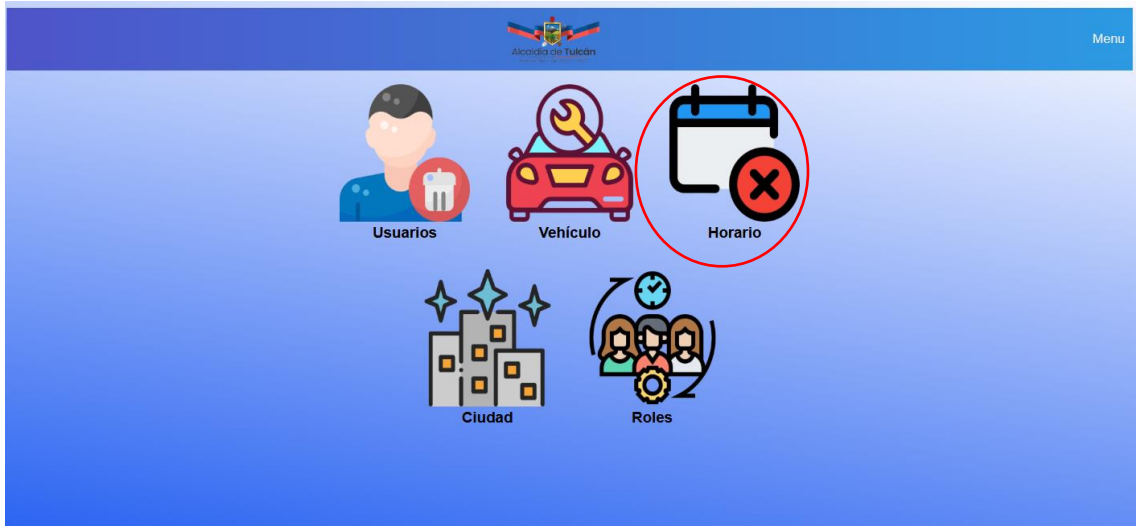


Figura 94. Pantalla eliminar horario

INFORMACIÓN

Seleccionamos ahora el módulo información, en donde se puede obtener información con respecto a visualizar en tablas usuarios, vehículo, roles, ciudad y horarios.



Figura 95. Pantalla principal información

menú de información, consta de 5 diferentes campos de datos registrados en nuestra base.

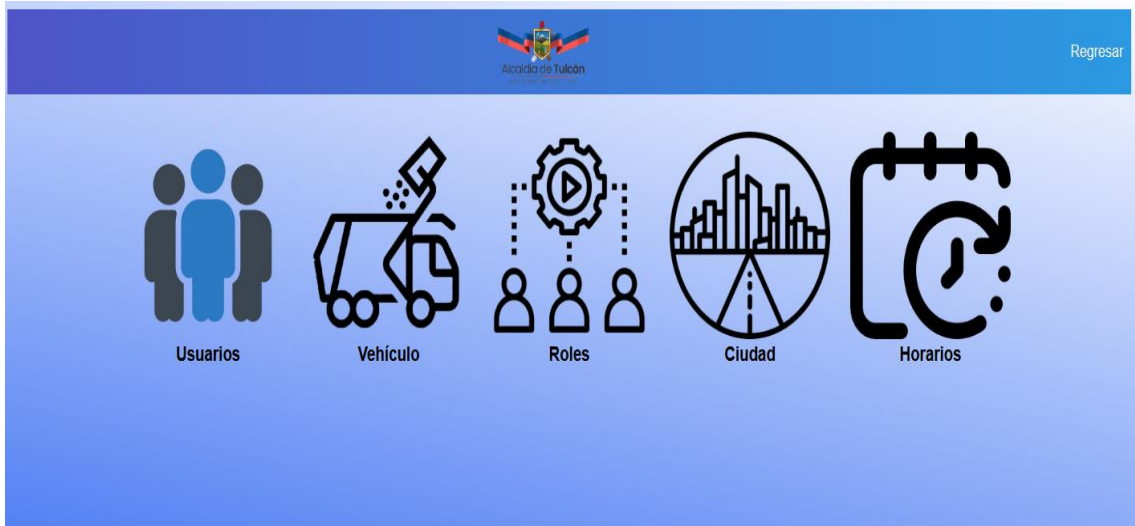


Figura 96. Pantalla información – menú de información

Información – Usuario

Seleccionamos el icono usuario para acceder a su venta.

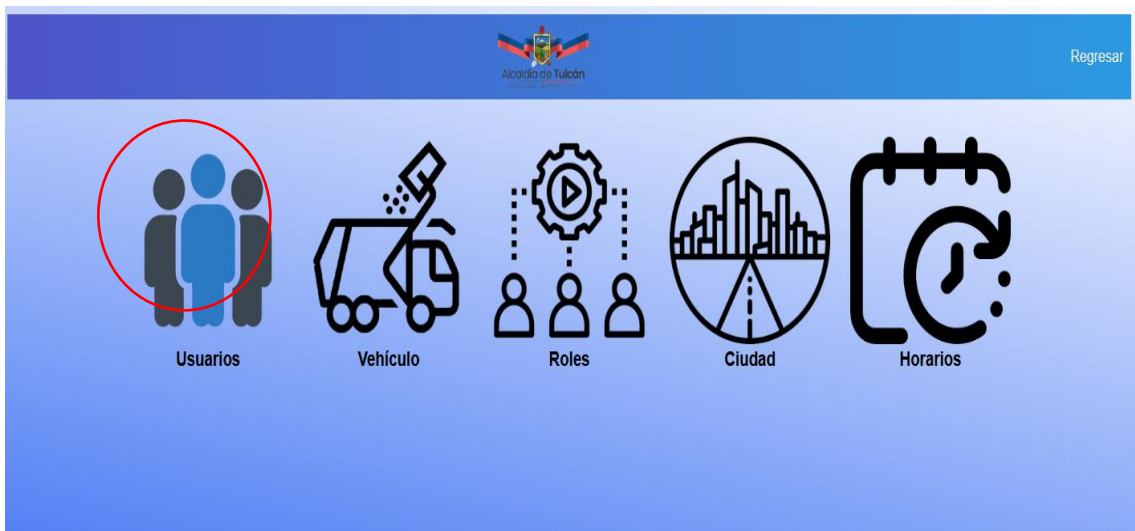


Figura 97. Pantalla información usuario – selección icono usuario

Dentro de la ventana información usuario esta la tabla de todos los registros de los usuarios con sus correspondientes datos; cédula, nombre, edad, ciudad, teléfono, correo y contraseña.

La contraseña está en estado encriptado por ser un dato delicado.

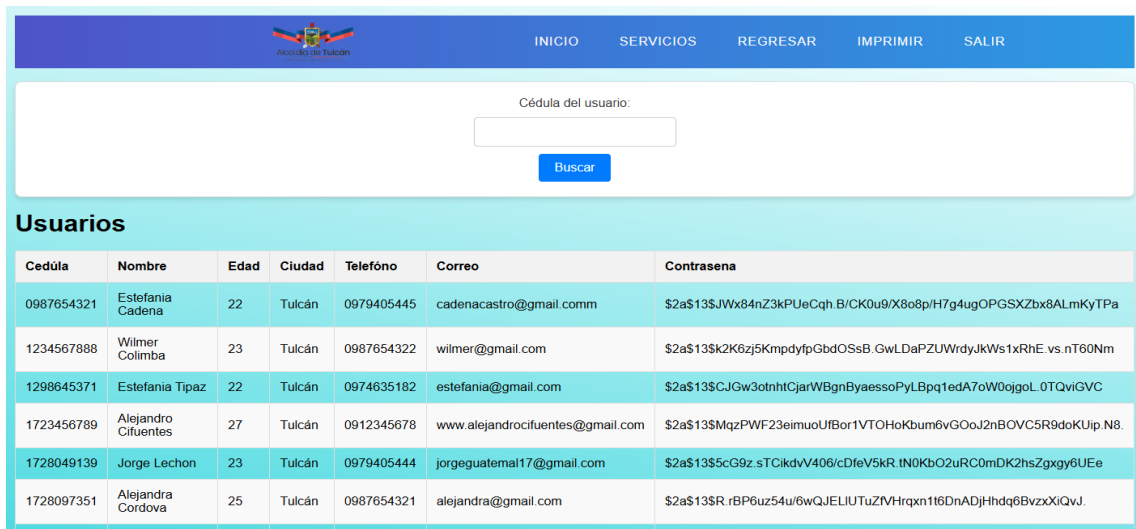


Figura 98. Pantalla información usuario

Esta presente un panel donde podemos hacer una búsqueda de algún usuario el comprobar su registro en la base de datos, también podemos realizar modificaciones en la información de dicho usuario.

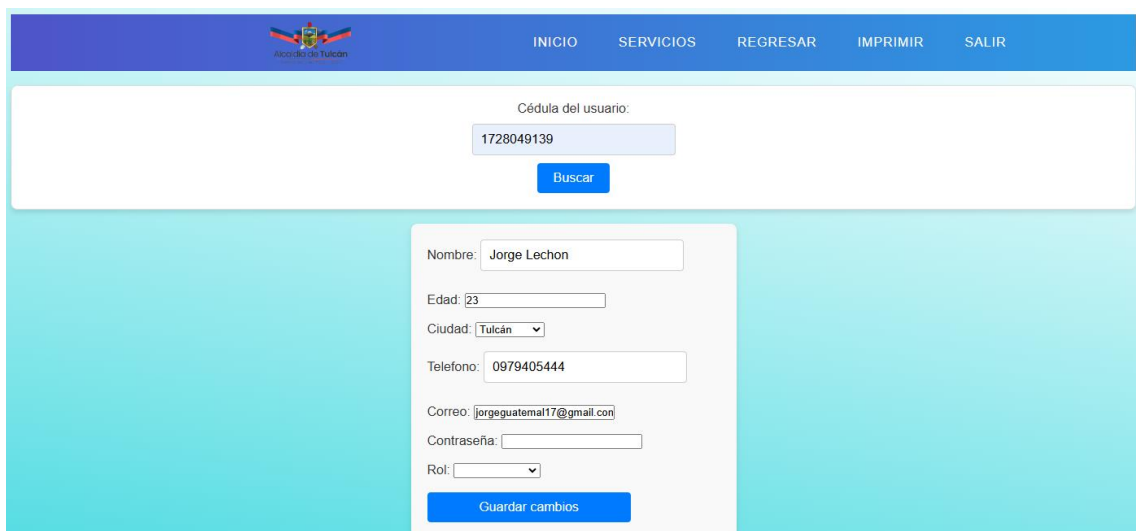


Figura 99. Pantalla información usuario

menú de navegación que cuenta esta ventana nos ofrece 5 opciones;

Inicio: Direcciona a la ventana principal del aplicativo.

Servicios: Presenta 3 opciones (agregar, eliminar y rutas) para agregar nos direcciona a la ventana de agregar usuarios y del mismo modo para eliminar a la ventana eliminar usuarios, estas opciones varían de acuerdo con la ventana que estemos navegando.

Regresar: Direcciona al menú información

Imprimir: Permite guardar e imprimir la información presentada.

Salir: Cierra sesión.



Figura 100. Pantalla información usuario

Información Vehículo

Seleccionamos el icono de vehículo para acceder a su respectiva ventana.



Figura 101. Pantalla información vehículo

Ventana información de vehículo consta de una tabla con todos los vehículos cargados a la base de datos.

Tenemos el menú de navegación con la misma funcionalidad que la anterior ventana de usuario, la parte que cambia es servicios que consta con sus 3 opciones (agregar, eliminar y rutas) la opción agregar direcciona a la ventana agregar vehículo y de la misma forma eliminar vehículo.

ID	Tipo	Marca	Codigo	Color	Año	Placa	Carga	Estado
1	Administrador	Internacional	05.01.13	Amarillo	2009	CMA-0134	Posterior	Operativo
2	Recolector	Internacional	05.01.14	Gris	2009	CMA-0133	Posterior	Dañado
3	Recolector	Hino	05.01.26	Blanco	2014	CMA-1028	Posterior	Operativo

Figura 102. Pantalla información vehículo

RUTAS

En el menú principal seleccionamos el módulo **rutas** para acceder.

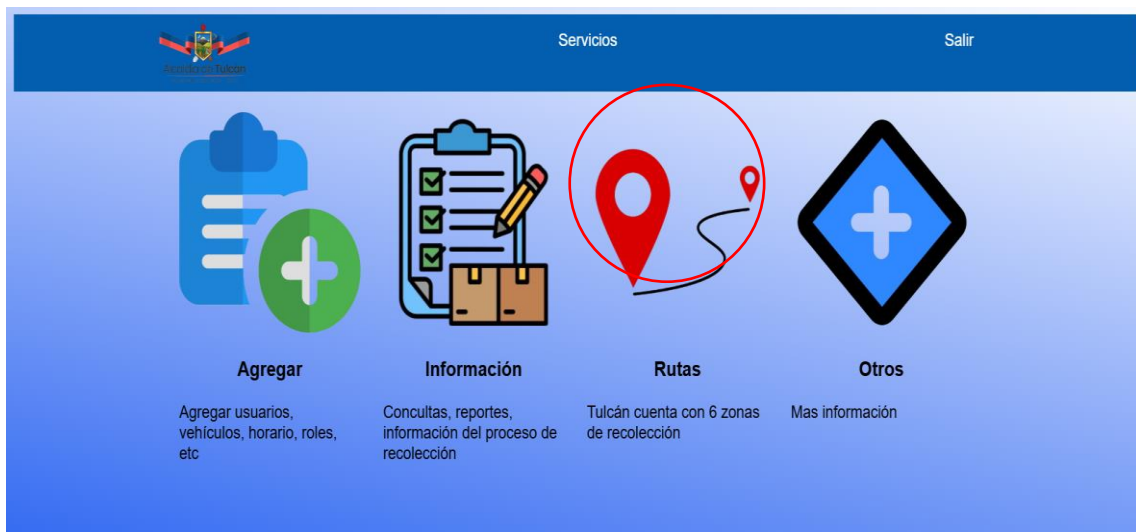


Figura 103. Pantalla menú principal

Dentro del módulo rutas podemos visualizar 6 iconos referentes a zonas, Tulcán cuenta con 5 zonas de recolección dentro de la ciudad y la zona test permite analizar, entender y manipular una zona.



Figura 104. Pantalla menú zona de recolección

Podemos seleccionar cualquier zona entre la 1 a 5 para planificar una ruta de recolección en este caso seleccionamos la zona 2.

Dentro de la ventana se puede visualizar botones, menú de navegación y un mapa que permite visualizar una zona específica de acuerdo con la seleccionada.

Tabla 32. Opciones desplegables

Menú navegación	<p>Inicio: Direcciona a la página menú principal del aplicativo.</p> <p>Zonas: Permite regresar al menú de zonas para seleccionar otra de las 5 planteadas</p> <p>Imprimir: Podemos guardar e imprimir la información que la página refleje.</p> <p>Salir: Cierra sesión del aplicativo.</p>
Mapa	<p>El mapa permite visualizar la zona específica de recolección, podemos ver puntos, rutas y de forma satelital el mapa de la ciudad.</p>
Información	<p>Cada zona de recolección cuenta con los datos del recorrido, como fecha, tiempo, chofer, auxiliares y el vehículo.</p>

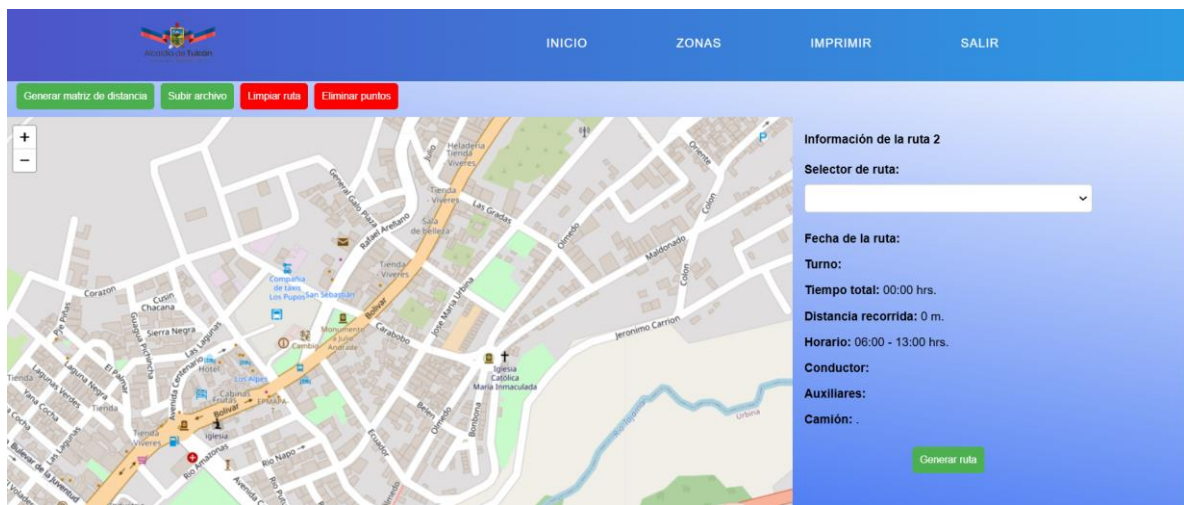


Figura 105. Pantalla zona – visualizar zona 2

BOTONES

Botón - subir archivo

Permite subir un archivo .xlsx es decir archivo un Excel de los puntos que referenciamos de una zona de la ciudad. Dicho archivo tiene un formato específico para que el aplicativo pueda leerlo.

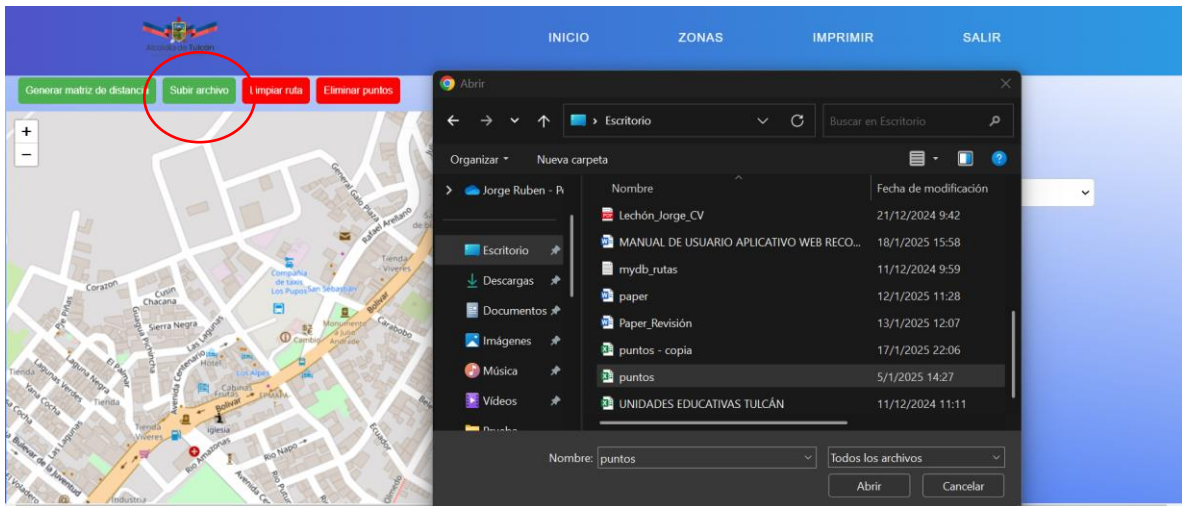


Figura 106. Pantalla zona – Zona 2

Archivo – Excel

El formato para subir los puntos al aplicativo web consta de 4 columnas.

Id: Es el nombre o referencia que va a tener el punto dentro del aplicativo.

Zona: Segmentamos los puntos para agruparlos de acuerdo con la zona que corresponda.

Latitud y longitud: Propiedad de coordenadas: 0.804838, -77.733814

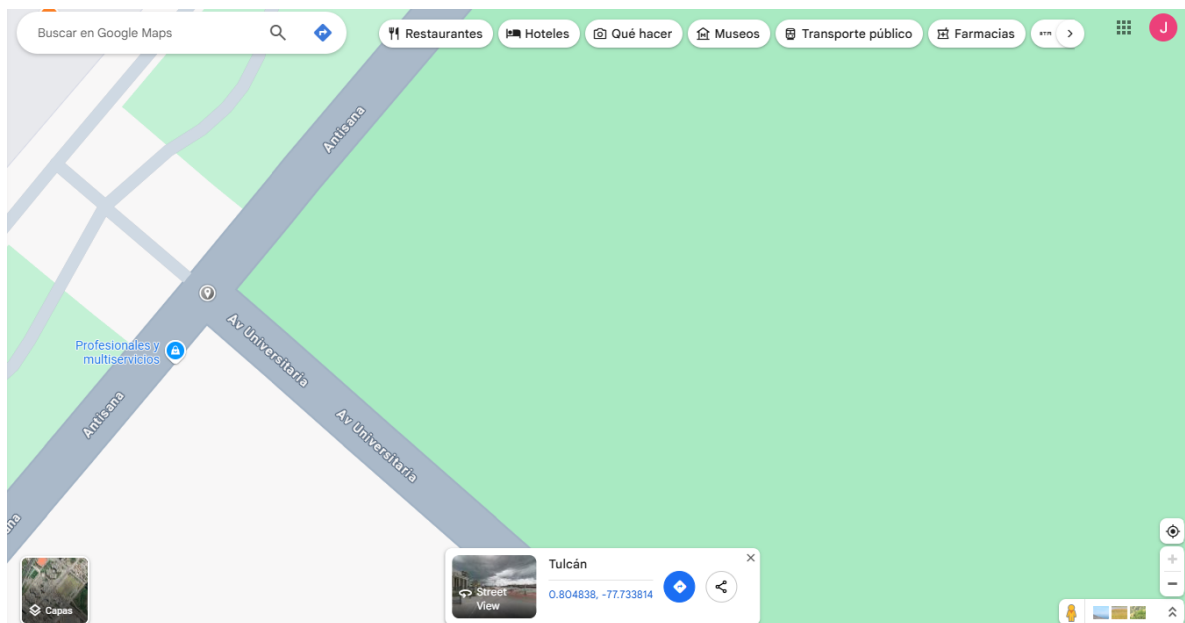


Figura 107. Pantalla Google maps con coordenadas

	A	B	C	D
1	id	zona	latitud	longitud
2	t1	test	0.808624,	-77.720869
3	t2	test	0.809356,	-77.719889
4	t3	test	0.808683,	-77.719366
5	t4	test	0.808034,	-77.718857
6	t5	test	0.809988,	-77.720442
7	t6	test	0.810668,	-77.720904
8	t7	test	0.811164,	-77.720284
9	t8	test	0.810511,	-77.719784
10	t9	test	0.809846,	-77.719275
11	t10	test	0.809205,	-77.718735
12	t11	test	0.809722,	-77.718083
13	t12	test	0.810367,	-77.718594
14	t13	test	0.811019,	-77.719125
15	t14	test	0.811711,	-77.719608
16	t15	test	0.812169,	-77.718980
17	t16	test	0.811540,	-77.718476
18	t17	test	0.810893,	-77.717945
19	t18	test	0.810236,	-77.717417
20	t19	test	0.810732,	-77.716764

Figura 108. Excel con puntos de recolección

Generar matriz de distancia

Esta función permite crear y guardar las relaciones entre los puntos referenciados en el mapa, crea una matriz distancia que está conectada con Neo4j. Este proceso es demoroso mientras más puntos contenga la zona mayor es la generación de sus relaciones.

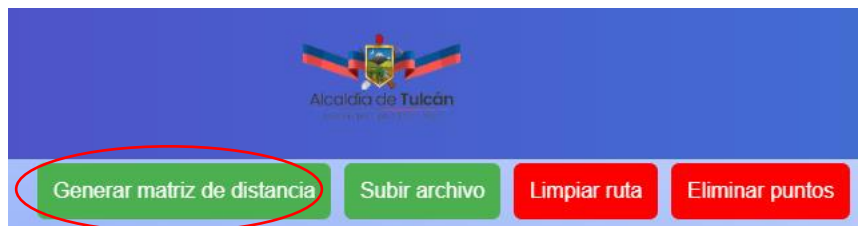


Figura 109. Pantalla zona – Botones zona 2

Limpiar registro

Elimina todos los registros de las zonas.

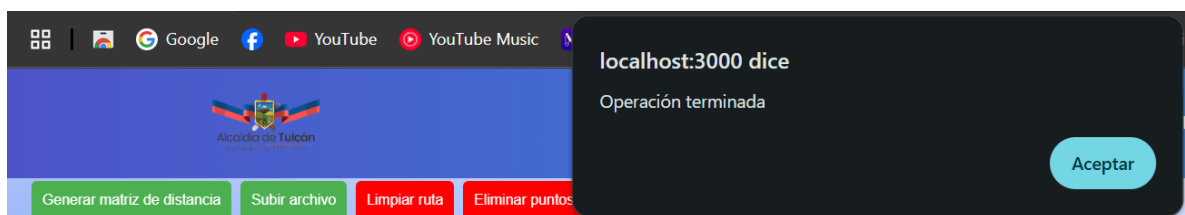


Figura 110. Pantalla zona – Limpiar registro de la zona 2

Registro de rutas vacío.

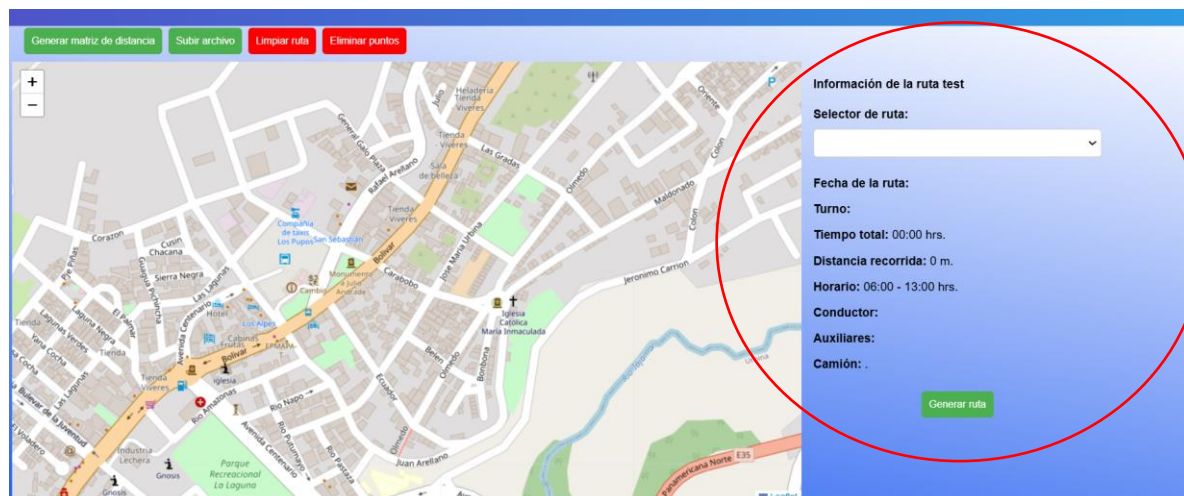


Figura 111. Pantalla zona – Registro de ruta vacío

Eliminar puntos

Esta función elimina los puntos referenciados anteriormente, en el caso de requerir cambiar, modificar una zona se debe limpiar los puntos propuestos anteriormente.

GENERAR RUTA

Permite generar una ruta para la recolección tomando en cuenta los parámetros que proponemos a los puntos ubicados en el mapa.

Al presionar el botón podemos visualizar una ventana en donde se debe seleccionar datos.

Tabla 33. Datos al generar una ruta

Hora inicio	Seleccionamos la hora en la que vamos a iniciar el servicio de recolección, permite llevar contabilidad de la hora de operación.
Camión	Tenemos las placas de los camiones que están en estado operativo, debemos seleccionar uno para continuar
Auxiliares	Esta cargado todas las personas que registran con el rol de auxiliar. Se puede seleccionar mínimo uno y máximo dos.
Tiempo de servicio	Tenemos definido en minutos el tiempo de servicio está en el rango de 1 minuto a 4 minutos.
Conductor	Aparecen todas las personas registradas con el rol de conductor se debe seleccionar uno.
Turno	Están presentes las jornadas laborales, matutino y vespertino.

Una vez completo los datos se debe pulsar el botón generar ruta, y en un tiempo corto comenzara a crear la ruta que el aplicativo propone.

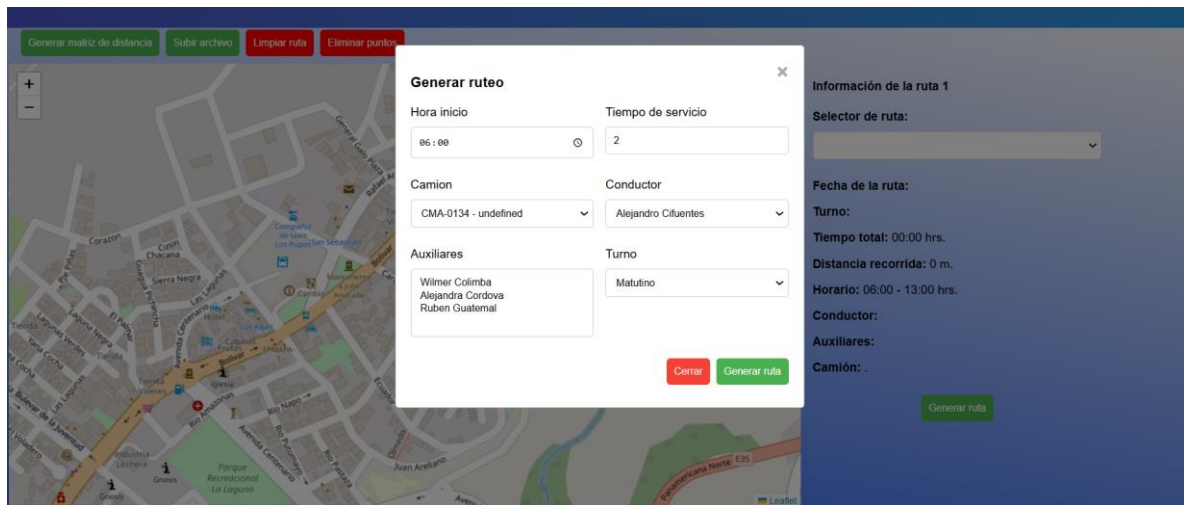


Figura 112. Pantalla zona – Generar ruta en la zona 2

Al generar ruta por primera vez el aplicativo muestra una ruta de forma aleatoria con todos los puntos de la zona.

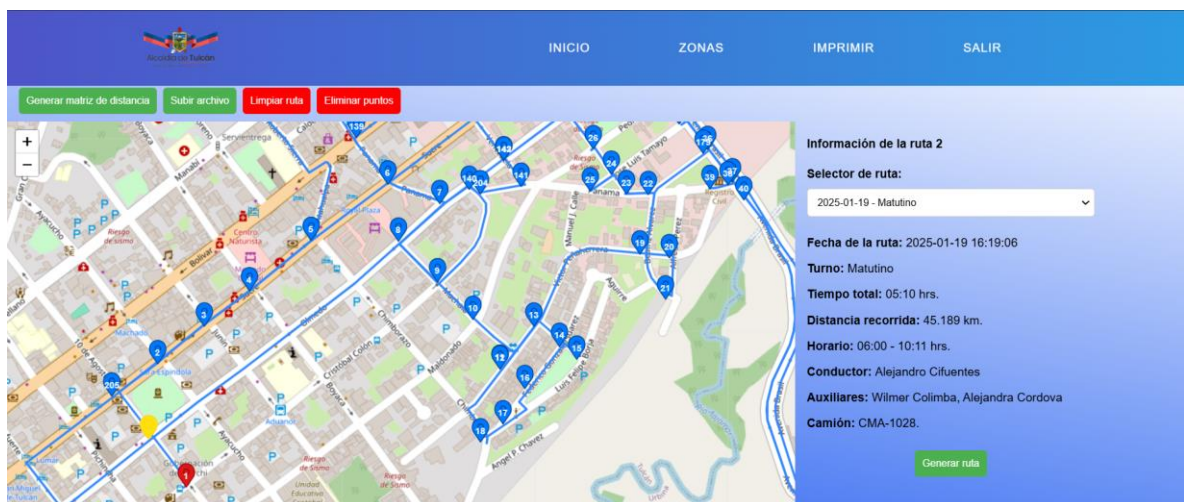


Figura 113. Pantalla zona 2

Para crear nuestra primera ruta de recorrido tenemos que establecer un punto de inicio y un punto final y eso podemos hacerlo manejando prioridades, al cargarse los puntos por primera vez todos los puntos tienen la misma prioridad de 1.

Prioridades

Las prioridades que maneja el aplicativo ayuda a cumplir una cobertura total de una zona, porque al tratarse de un sistema de recolección de desechos sólidos la cobertura total es primordial.

El rango de prioridad va desde 0 hasta el 20, entre menor sea la prioridad mayor será el objetivo de llegar a ese punto primero, la prioridad 0 se puede utilizar para definir un punto de partida. Y entre mayor sea la prioridad será el punto último en tomarse en cuenta.

Ejemplo:

Punto uno tiene una prioridad de 0, el punto 40 tiene una prioridad de 1 y el punto 220 tiene la prioridad de 20. Entonces el punto uno sería un punto de partida y el punto 220 sería el punto final del recorrido.

Para establecer una nueva prioridad pulsamos el punto que vamos a cambiar.

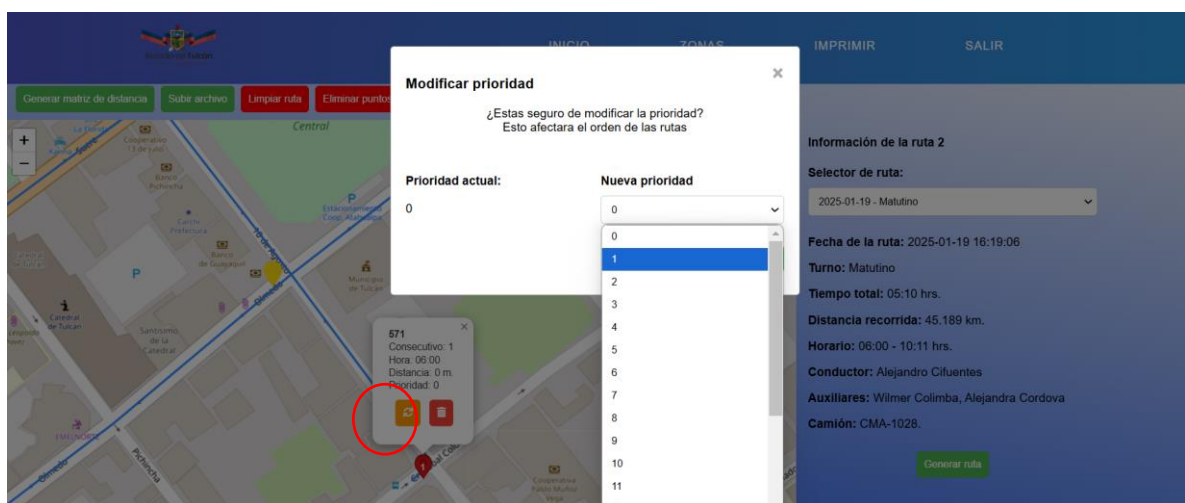


Figura 114. Pantalla zona – Prioridad en el punto 1 de la zona 2

Además, que podemos eliminar los puntos de la zona, de la misma forma pulsando sobre el punto que queremos eliminar.

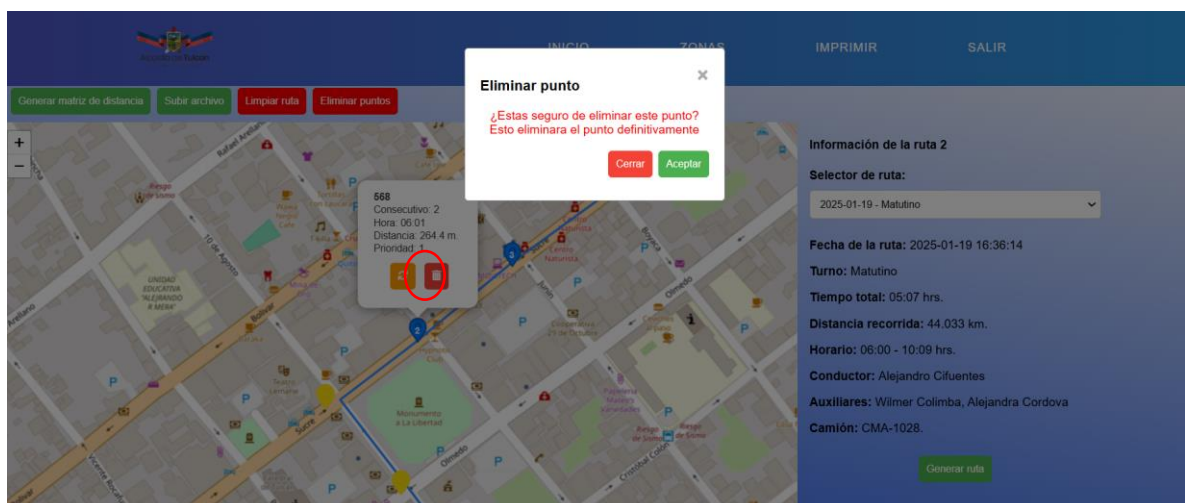


Figura 115. Pantalla zona – Eliminar punto 2 de la zona 2

Una vez ya con modificaciones y parámetros a los puntos se puede generar la nueva ruta.

Cuando la ruta esta creada, debemos seleccionar en el buscador "selector de ruta" y la ruta creada reciente estará ubicada en la última opción, además de poder verificarlo con la fecha y hora de creación.

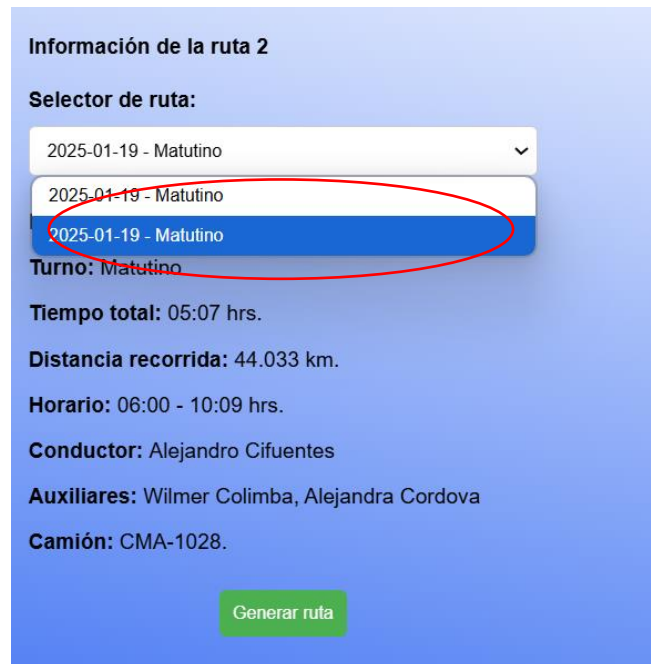


Figura 116. Pantalla zona – Visualización zona 2

Zona 2, ruta creada

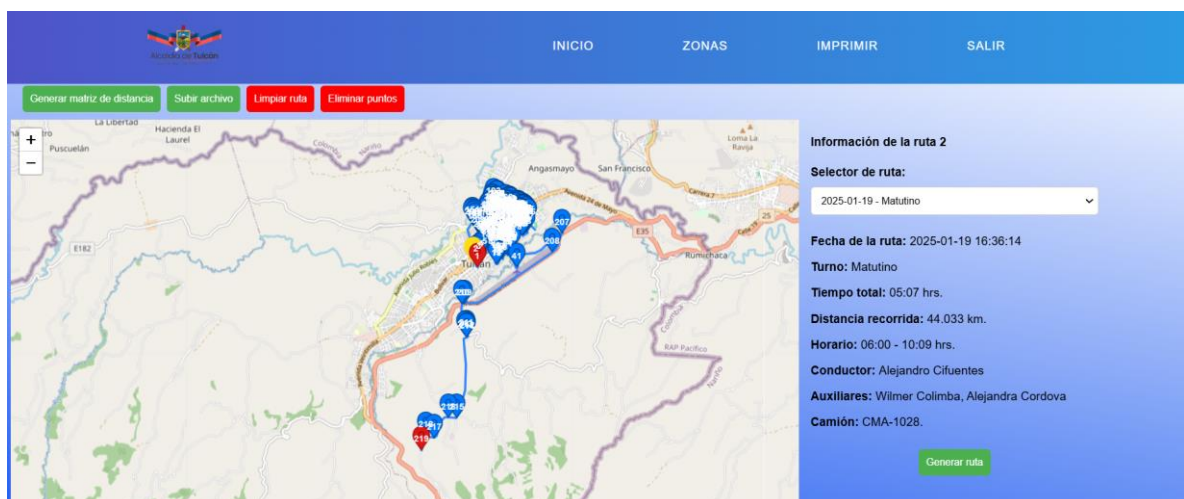


Figura 117. Zona 2

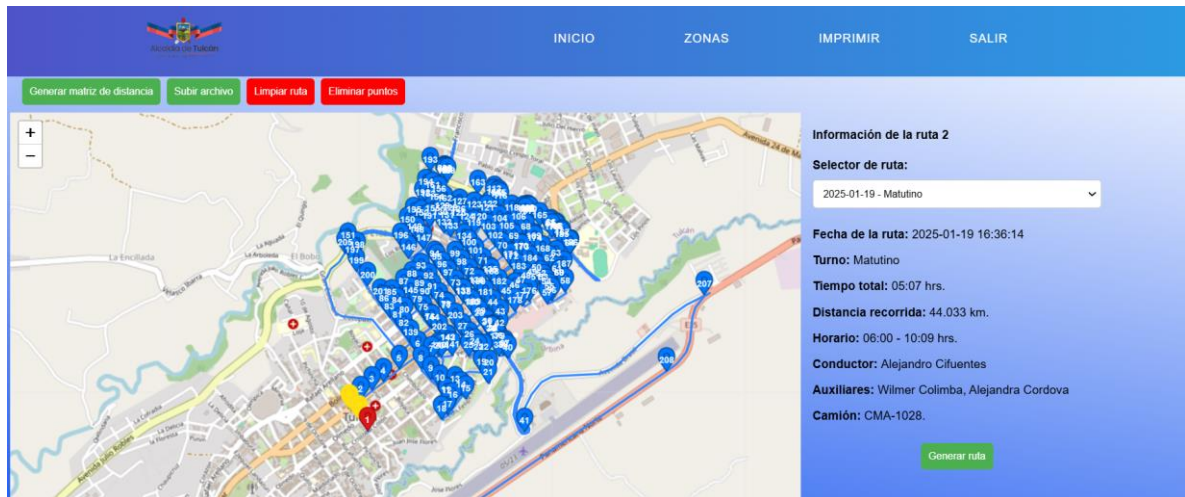


Figura 118. Puntos de recolección Zona 2

Los puntos, contiene información tales como nombre del punto, la secuencia en forma numérica la que se debe seguir para recorrer, distancia acumulada hasta ese punto y por último la prioridad establecida.

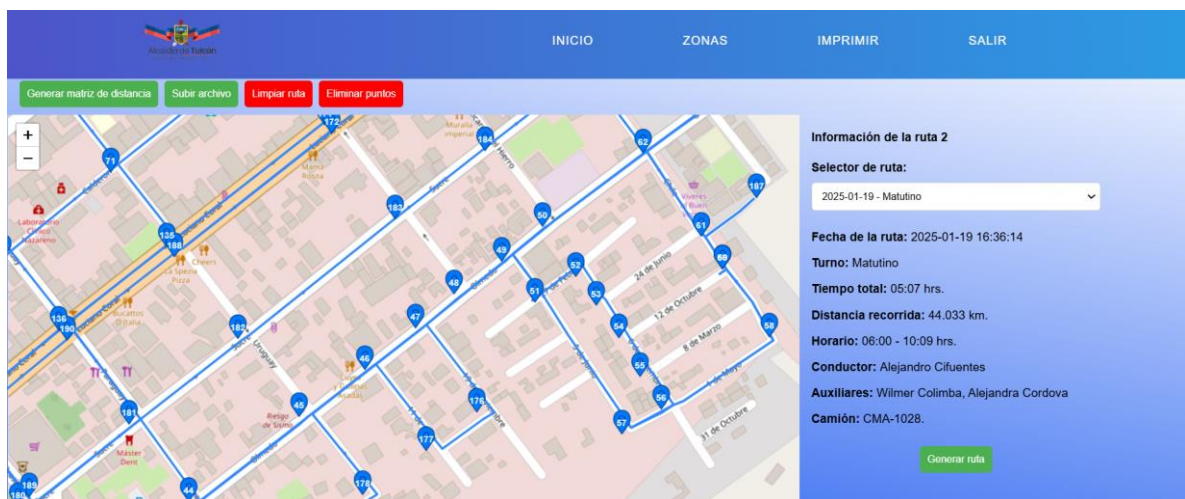


Figura 119. Pantalla del recorrido de la zona 2

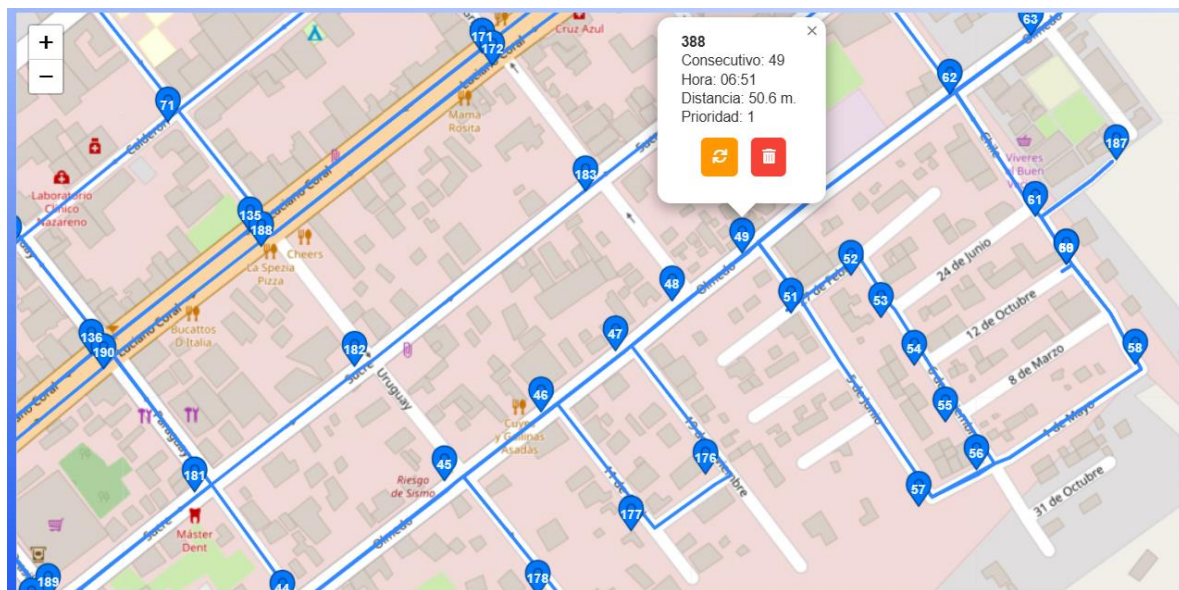


Figura 120. Visualización de puntos de zona 2

Anexo 12. Manual Técnico

UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI
FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES



CARRERA DE COMPUTACIÓN

Manual Técnico

GAD Municipal de Tulcán

Instituto de Ambientes

Versión: 1.0

Fecha: mayo 2024

DISEÑO SISTEMA RUTAS ÓPTIMAS PARA LA RECOLECCIÓN DE DESECHOS SÓLIDOS

Términos utilizados para el diseño y codificación

Tabla 34: Términos técnicos

Termino	Descripción
Mydb_rutas	Base de datos para almacenar la información sobre la recolección de desechos sólidos.
Server.js	Archivo esencial dentro del aplicativo, actúa como servidor principal para gestionar las solicitudes HTTP basada en Express
Express	Marco para gestionar solicitudes HTTP
Main.js	Archivo que realiza ejecución de scripts como; limpiar datos, recargar, función de redirección, API.
Auth.js	Usuarios registrados para establecer roles, scripts que permite administrar entre un usuario y administrador.
Index.html	Actúa como la pantalla principal del aplicativo.
api	Es un servicio que permite obtener información sobre rutas entre dos o más ubicaciones.
Axios	Librería para crear instancias personalizadas de cliente HTTP que interactúa con el api de OSRM (Open Source Routing Machine).
Token-Privado/público	Valida el token de autenticación y el rol de usuario en el aplicativo.
Package.json	Listas de dependencias necesarias para el aplicativo.
MVC	Modelo, Vista y Controlador

Estructura de módulos

ESTÁNDAR DE CODIFICACIÓN

Código JavaScript

Directorios

\js (Todos los scripts de javascript)

\css (Todas las hojas de estilos)

\img (Todas las imagenes)

\HTML(Páginas web)

SERVIDOR PRINCIPAL DEL APLICATIVO Y CONEXIÓN CON LA BASE DE DATOS - SERVER.JS

Express configurado con un servidor web para poder interactuar con una base de datos, base de datos utilizado

```
mysql.const express = require('express');
```

```

const bodyParser = require('body-parser');
const { encriptarContraseña } = require('./src/helpers/authHelper');
const app = express();
const port = 3000;
const routes = require('./src/modules/index');
routes(app)
// Configuración de la conexión a la base de datos
const { connection, queryAsync } = require('./src/conexion')
// Middleware para parsear el cuerpo de las solicitudes
app.use(bodyParser.json());
// Endpoint para obtener datos de la base de datos
app.get('/datos', (req, res) => {
    connection.query('SELECT * FROM ciudad', (err, results) => {
        if (err) {
            console.error('Error ejecutando la consulta:', err);
            res.status(500).send('Error en la consulta');
            return;
        }
        res.json(results);
    });
});
// Servir archivos estáticos (tu frontend)
app.use(express.static('public'));
app.listen(port, () => {
    console.log(`Servidor corriendo en http://localhost:${port}`);
});

```

Tabla 35: Interpretación de código

Código	Descripción
Express	configura un servidor y maneja las rutas
Neo4j	Maneja puntos geográficos
getpointsByZoneNEeo4j	Maneja puntos geográficos
setPointNeo4j	Crea un punto en Neo4j
getPuntosByZona	Obtiene puntos de otra base de datos, organizados por zonas
getPuntosOrdenados	Recupera puntos ordenados

CREACION DE RUTAS – RUTAS.JS

Extrae el valor zona del cuerpo de la solicitud, este valor identifica la zona geográfica que se procesa.

```
const { zona } = req.body
```

Función `getPuntosByZona(zona)` recupera ubicaciones es decir puntos de una zona especificada desde la base de datos

```
const points = await getPuntosByZona(zona)
```

Compara los puntos obtenidos con los datos almacenados de Neo4j, posiblemente para sincronizar la base de datos o en si asegurarse que los puntos existen en Neo4j.

```
await comparePointsNeo4j(points, zona)
```

Recupera relaciones, conexión entre puntos existentes para la zona en la base de datos Neo4j.

```
const relationsPoints = await getRelationPoints-ByZone(zona, 'OSM')
```

Double bucle, recorre cada par de puntos punto1, punto 2 dentro de los puntos de la zona.

Si el punto 1 y el punto 2 son iguales el id salta al siguiente para continuar con el proceso.

```
for(const point1 of points) {  
  for(const point2 of points) {  
    if(point1.id === point2.id) continue
```

Verifica si existe ya una relación entre el punto1 y punto2, de existir omite el par y continua.

```
const isRelation = relationsPoints.some(pointFind => pointFind.start.id ==
point1.id && pointFind.end.id == point2.id)

    if(isRelation) continue
```

Se realiza una solicitud GET en la API OpenStreetMap para calcular la ruta entre las coordenadas punto1 y punto2.

Parámetro overview: "full" puede indicar que se quiere una representación detallada de la ruta.

```
const { data } = await
osmRouteApi.get(`/ ${point1.longitud}, ${point1.latitud}; ${point2.longitud}, $
{point2.latitud}`, {
    params: {
        overview: 'full'
    }
}
```

Si la API no devuelve rutas, salta al siguiente par.

Extrae información: distance, duration y geometry

```
const { routes } = data
if(routes.length === 0) continue
const { distance, duration, geometry } = routes[0]
```

Usa la función setRelationPointNeo4j guarda la relación entre el punto1 y punto2 en la base de datos Neo4j, junto con los datos; distance, duration y geometry

```
await setRelationPointNeo4j({
    idStart: point1.id,
    idEnd: point2.id,
    distance,
    time: duration,
    codedPolylines: geometry
}, 'OSM')
```

Imprime en consola las coordenadas de los puntos y los datos calculados

```
console.log(` punto1:  ${point1.latitud},${point1.longitud}
punto2: ${point2.latitud},${point2.longitud}` )

console.log(` distancia:      ${distance}      dura-cion:
${duration}` )
```

Al terminar de procesar todos los puntos y relaciones, envía un mensaje a JSON indicando cuántos puntos fueron utilizados para crear rutas

```
res.json({ message: `Rutas creadas con: ${points.length} puntos` })
})
```

CREACIÓN DE RUTAS – RUTAS.JS

Se implementa en la ruta POST/create en un servidor Express. Tiene como proceso automatizar la creación de rutas logísticas dentro de una zona específica, gestionando puntos, tiempos, distancias y relaciones.

RECEPCIÓN DE DATOS DESDE EL CLIENTE

Extraer datos del body de la solicitud (req.body):

Código	Descripción
Zone	Zona geográfica donde se generarán las rutas.
notPoints	Lista de puntos a excluir de las rutas.
startTime	Hora de inicio del viaje.
timeService	Tiempo estimado del servicio por punto.
Drive	Identificador del camión.
Auxs	Lista de auxiliares (personal adicional).
Turn	Turno del viaje.

```
const {zone, notPoints, startTime, timeService, driver, truck, auxs, turn} = req.body;
```

OBTENCIÓN DEL PUNTO INICIAL

Usa `getPuntosOrdenados(zone)` para obtener los puntos de la zona, organizados.

Selecciona el primer punto (`startPoint`) como el punto de partida del viaje.

```
const startPoint = await getPuntosOrdenados(zone);
```

```
let startPointObj = startPoint[0];
```

FILTRADO DE PUNTOS NO VÁLIDOS

pointsNotFound: incluye el id del punto inicial y los puntos a excluir (notPoints)

Usa `getPuntosByZonaNoIds(zone, pointsNptFound)` para obtener los puntos válidos dentro de la zona que no están en `pointsNotFound`.

```
const pointsNotFound = [startPointObj.id, ...notPoints];
```

```
const points = await getPuntosByZonaNoIds(String(zone), pointsNotFound);
```

Obtiene las relaciones (conexiones entre puntos) desde la base de datos de Neo4j.

```
const relationsPoints = await getRelationPoints-ByZone(String(zone), 'OSM');
```

PREPARACIÓN DE DATOS

Se hace una copia profunda de los puntos para evitar problemas al manipularlos durante la generación de rutas.

```
const allPoints = JSON.parse(JSON.stringify(points));
```

GENERACIÓN DE RUTAS

Usa la función `route` para generar las rutas:

Puntos activos (`status === 1`): filtra puntos válidos para el viaje.

Relaciones existentes (`relationsPoints`): Calcula las rutas usando datos previos de la base de datos.

Punto Inicial (`startPointObj`): Define el punto de partida.

```
const routes = route (  
  points.filter(point => point.status === 1),  
  allPoints,  
  relationsPoints,  
  [{ point: startPointObj, properties: {} }]  
);
```

PREPARACIÓN DE RUTAS PARA LA BASE DE DATOS

Crear un arreglo `routesInsert` con las rutas procesadas:

Hora_inicio y hora_fin: Calcula los horarios de inicio y fin para cada punto de la ruta.

Servicio: Convierte el tiempo de servicio en segundos.

Distancia, tiempo y polylinea: Usa datos de la ruta generada por route.

fk_punto: Vincula cada ruta al punto correspondiente.

Manejo de tiempos consecutivos

Calcula hora_inicio sumando la duración del viaje previo a hora_fin de la ruta anterior.

```
const routesInsert = [];  
let consecutive = 1;  
for(const {properties, point} of routes) {  
  if(!point) continue;  
  const {distance, time, codedPolylines} = properties;  
  let startTimeRoute = '';  
  let endTimeRoute = '';  
  const service = timeService * 60;  
  if(consecutive === 1) {  
    startTimeRoute = startTime;  
  } else {  
    const lastRoute = routesInsert[consecutive - 2];  
    startTimeRoute = addSecondsTo-Time(lastRoute.hora_fin, time || 0);  
  }  
  endTimeRoute = addSecondsToTime(startTimeRoute, service);  
  routesInsert.push({  
    consecutivo: consecutive,  
    hora_inicio: startTimeRoute,  
    hora_fin: endTimeRoute,  
    servicio: service,  
  });  
}
```

```

    distancia: distance || 0,
    tiempo: time || 0,
    polyline: codedPolylines || '',
    fk_punto: point.id
  });
  consecutive++;
}

```

INSERCIÓN DE RUTAS EN LA BASE DE DATOS

Guardar las rutas procesadas en la base de datos usando la función `setRutas`.

`routesDB` contiene los datos insertados, incluidos sus identificadores únicos.

```
const routesDB = await setRutas(routesInsert);
```

Cálculo de Totales

Distancia total (`totalDistance`).

Tiempo total de viaje (`totalTimeTravel`).

Tiempo total de servicio (`totalTimeService`).

Extrae los ids de las rutas insertadas (`routesids`).

```
const totalDistance = routesDB.reduce((acc, route) => acc + route.distancia,
0);
```

```
const totalTimeTravel = routesDB.reduce((acc, route) => acc + route.tiempo,
0);
```

```
const totalTimeService = routesDB.reduce((acc, route) => acc +
route.servicio, 0);
```

```
const routesIds = routesDB.map(route => route.id);
```

CREACIÓN DEL VIAJE

Registra un viaje en la base de datos con, totales calculados, zona, conductor, camión y turnos asignados.

```
const viaje = await setViaje({
  distanciaTotal: totalDistance,
```

```

    viajeTotal: totalTimeTravel,
    servicioTotal: totalTimeService,
    zona: zone,
    conductor: driver,
    camion: truck,
    auxs: auxs.join(', '),
    turno: turn
  });

```

RELACIÓN ENTRE VIAJE Y RUTAS

Establece una relación entre el viaje creado y las rutas generadas en la base de datos.

```
await setViajeRutasRel(viaje.insertId, routesIds);
```

RESPUESTA CLIENTE

Envía una respuesta confirmando que el proceso se completó con éxito.

```
res.json({ message: "Ruta creada correctamente" });
```

ALGORITMO

Tabla 37: Respuesta Cliente

Código	Descripción
Points	Una lista de puntos que aún no han sido visitados
allPoints	Todos los puntos disponibles (se usa para contexto general, posiblemente para búsquedas globales).
Relations	Una lista de relaciones entre puntos, que incluye información como el punto inicial, el punto final y sus propiedades (distancia, tiempo).
Routes	La lista actual de puntos visitados y sus propiedades correspondientes (la ruta construida hasta el momento).

```
const route = (points, allPoints, relations, routes) => { ... }
```

FLUJO DE CÓDIGO

Si ya no hay puntos para visitar (puntos vacíos) la función termina y devuelve la ruta construida (routes).

```
if(points.length === 0) {
```

```
    return routes;
}
```

OBTENER EL ÚLTIMO PUNTO DE LA RUTA ACTUAL

Encuentra el último punto agregado a la ruta para continuar desde allí.

```
const ultPoint = routes[routes.length - 1].point;
```

Filtrar relaciones relevantes

Busca todas las relaciones que parten desde último punto (ultPoint).

```
const pointStarts = relations.filter((point) => point.start.id ===
ultPoint.id);
```

BUSCAR LA RELACIÓN MÁS CORTA

Llama a una función auxiliar getShortesRoute que encuentra la relación más corta desde el punto actual hacia un punto disponible en puntos.

```
const nextPointNeo4j = getShortestRoute(pointStarts, points, allPoints,
ultPoint);
```

ENCONTRAR LA RELACIÓN

Busca una relación válida (nextPointNeo4j)

-Encuentra el índice del siguiente punto en la lista de puntos disponibles.

-Agrega el siguiente punto a la ruta actual (routes) con sus propiedades; distancia y tiempo

-Elimina el siguiente punto de la lista points para marcarlo como visitado

```
if(nextPointNeo4j) {
    const nextPointIndex = points.findIndex(point => point.id ===
nextPointNeo4j.end.id);
    routes.push({
        point: points[nextPointIndex],
        properties: nextPointNeo4j.properties
    });
    removeObjectToArr(points, points[nextPointIndex]);
```

```
}
```

BÚSQUEDA NO ENCUENTRA UNA RELACIÓN VALIDA

Al no encontrar una relación válida desde el punto actual, imprime un mensaje de error, elimina un punto arbitrario de la lista de puntos.

```
return route(points, allPoints, relations, routes);
```

LLAMADA RECURSIVA

Llama la función nuevamente con las listas actualizadas de puntos, relaciones y rutas

```
return route(points, allPoints, relations, routes);
```

FUNCIÓN AUXILIARES

`getShortesRoute`: Encuentra la relación más corta entre el punto actual y los puntos disponibles, utiliza propiedades como distancia o tiempo para tomar una decisión.

`removeObjectToArr`: Elimina un objeto específico de un arreglo, asegura que los puntos visitados no se vuelvan a considerar.

PRIORIDADES – RUTAS.JS

FUNCIÓN

`Relations`: Lista de relaciones posibles desde un punto dado. Cada relación incluye un punto de inicio (`start`), un punto de fin (`end`) y propiedades como distancia

`Points`: Lista de puntos disponibles que aún no se han visitado.

`allPoints`: Todos los puntos del sistema, con información completa, verifica detalles adicionales de prioridades.

`Point`: El punto actual desde el que están evaluando las relaciones.

```
const getShortestRoute = (relations, points, allPoints, point) => { ... }
```

FLUJO DE CÓDIGO

Busca inicialmente una relación desde el punto actual hacia el primer punto de la lista `points`.

Si no se encuentra ninguna relación válida, la función termina devolviendo `undefined`.

```
let distanceObj = undefined;
```

```
distanceObj = relations.find((relation) => relation.end.id ===  
points[0].id);
```

```
if (!distanceObj) return undefined;
```

ITERAR SOBRE TODAS LAS RELACIONES

Recorre todas las relaciones para encontrar la mejor opción según las condiciones establecidas

```
for (const relation of relations) {  
  ...  
}
```

Verifica si el punto final de la relación está en points

Comprueba si el punto final de la relación actual está en la lista de puntos aún disponibles (points).

Si no está, pasa a la siguiente relación.

```
const isExist = points.some(pointF => pointF.id === relation.end.id);
```

```
if (!isExist) continue;
```

OBTENER INFORMACIÓN COMPLETA DEL PUNTO FINAL

Busca el punto final en allPoints para obtener información adicional, como su prioridad

Si no encuentra información sobre el punto, pasa a la siguiente relación.

```
const pointRelation = allPoints.find(pointF => pointF.id ===  
relation.end.id);
```

```
if (!pointRelation) continue;
```

DETERMINAR PRIORIDAD

Compara la prioridad del primer punto en points con la prioridad del punto final de la relación actual.

Si la prioridad del punto en points es mayor o igual, se considera prioritario.

```
let isPriority = (points[0].prioridad || 0) >= (pointRelation.prioridad ||  
0);
```

VALIDAR Y CONVERTIR LAS DISTANCIAS

Comprueba que las prioridades de la relación contengan una distancia válida

Convierte la distancia a un número para asegurar que se puede comparar.

```
if (!relation.properties || isNaN(Number(relation.properties.distance)))  
  continue;
```

```
const relationDistance = Number(relation.properties.distance);
```

Comparar distancias y actualizar distanceObj

Si la relación actual tiene mayor prioridad y una distancia más corta que la relación guardada en distanceObj, actualiza distanceObj con la nueva relación

```
if (isPriority && relationDistance < distanceObj.properties.distance) {  
  distanceObj = relation;  
}
```

DEVOLVER LA MEJOR RELACIÓN

Devuelve la relación con la menor distancia, que además cumpla las condiciones de prioridad y validez.

```
return distanceObj;
```