



# **Manual de Usuario**

## **Routing App**

Aplicación para gestión de rutas TSP

Gabriela Dayana Coaquira Suyo  
Rafael Diego Nina Calizaya  
Dennis Javier Quispe Saavedra  
Christian Henry Venero Guevara  
Alexander Villafuerte Quispe

Universidad San Agustín, Arequipa  
Tecnología de objetos

03 de diciembre de 2025

## Índice

<b>1. Introducción.....</b>	<b>3</b>
1.1. Audiencia Objetivo.....	3
<b>2. Requisitos del Sistema.....</b>	<b>3</b>
2.1. Hardware.....	3
2.2. Software.....	3
<b>3. Instalación.....</b>	<b>3</b>
3.1. Windows.....	3
3.2. Linux.....	3
<b>4. Descripción de la Interfaz.....</b>	<b>3</b>
4.1. Panel Izquierdo de Control.....	4
4.1.1. Configuración de Operación.....	4
4.1.2. Algoritmos.....	4
4.1.3. Selección Manual de Nodos.....	4
4.1.4. Opciones TSP.....	5
4.1.5. Acciones Principales.....	5
4.2. Panel Derecho: Área de Visualización.....	5
4.3. Panel inferior: Resultados.....	5
<b>5. Gestión de Nodos.....</b>	<b>6</b>
5.1. Modos de selección.....	6
5.2. Definición del nodo inicio.....	6
5.3. Definición de los nodos destino.....	6
5.4. Botones de Control.....	6
<b>6. Características Adicionales.....</b>	<b>7</b>
6.1. Perfil de vehículo.....	7
<b>7. Casos de Uso.....</b>	<b>7</b>
7.1. Caso 1: Planificación de una ruta cerrada (TSP Estándar).....	7
7.2. Caso 2: Planificación de una ruta abierta (Path TSP).....	8
7.3. Caso 3: Cálculo de la ruta corta entre dos puntos.....	8
7.4. Caso 4: Limpieza y reconfiguración de la entrada.....	8
<b>8. Preguntas Frecuentes.....</b>	<b>9</b>
<b>9. Solución de Problemas.....</b>	<b>9</b>
9.1. La aplicación no inicia.....	9
9.2. No aparece el mapa.....	9
9.3. Error al ejecutar algoritmos.....	9
<b>10. Glosario.....</b>	<b>9</b>
<b>11. Contacto y Soporte.....</b>	<b>9</b>

## 1. Introducción

En la era digital moderna, los sistemas de enrutamiento geográfico son críticos para la eficiencia logística. En ciudades intermedias como Arequipa, el crecimiento desordenado del parque automotor ha incrementado los tiempos de traslado, afectando la productividad de las micro y pequeñas empresas MYPES. Estas organizaciones requieren herramientas que no solo encuentren la ruta más corta entre dos puntos, sino que optimicen secuencias complejas de entrega.

Este proyecto presenta una solución de software que integra algoritmos clásicos y heurísticas avanzadas sobre la red vial real de la ciudad. A diferencia de soluciones comerciales, nuestra propuesta permite la personalización de perfiles de vehículo y funciona de manera offline, evaluando el desempeño de técnicas de optimización computacional frente a las limitaciones de infraestructura tradicionales.

### 1.1. Audiencia Objetivo

Este manual es principalmente dirigido hacia personal operativo y directivo de Micro y Pequeñas Empresas (MYPES) en ciudades intermedias, que buscan una solución de enrutamiento avanzada, personalizable y funcional sin conexión a internet, para mitigar el impacto del tráfico desordenado en sus procesos logísticos.

## 2. Requisitos del Sistema

### 2.1. Hardware

#### Requisitos Mínimos:

- Procesador: CPU de doble núcleo a 2.0 GHz o superior
- Memoria RAM: 2 GB mínimo (4 GB recomendado)
- Espacio en disco: 200 MB para la aplicación + 50 MB para datos del grafo
- Pantalla: Resolución mínima de 1024x768 píxeles
- Dispositivo señalador: Mouse o touchpad para interacción con el mapa

#### Requisitos Recomendados:

- Procesador: CPU de cuatro núcleos a 2.5 GHz o superior
- Memoria RAM: 8 GB o más (para grafos grandes de 50k+ nodos)
- Espacio en disco: 500 MB libres
- Pantalla: Resolución de 1920x1080 o superior para mejor experiencia visual
- Tarjeta gráfica: GPU con soporte OpenGL 2.0 o superior (para renderizado fluido)

### 2.2. Software

#### Sistemas Operativos Soportados:

- **Windows:** Windows 10 (64-bit) o superior
- **Linux:** Ubuntu 20.04 LTS o superior, Fedora 34+, Debian 11+

- **macOS:** macOS 10.15 (Catalina) o superior

#### **Dependencias Requeridas:**

- **Qt Framework:** Versión 5.15+ o 6.0+ (Qt Widgets, Qt Core, Qt Gui)
- **Compilador C++17:**
  - **Windows:** Microsoft Visual C++ 2022 o MinGW-w64
  - **Linux:** GCC 11+ o Clang 14+
  - **macOS:** Xcode 12+ (incluye Clang 14+)
- **CMake:** Versión 3.16 o superior

#### **Bibliotecas del Sistema:**

OpenGL 2.0 o superior (generalmente incluido en los sistemas operativos modernos)  
Bibliotecas estándar de C++ (incluidas con el compilador)

### **3. Instalación**

#### **3.1. Windows**

##### Instalación desde Ejecutable Precompilado

1. Descargue el archivo .zip del release más reciente desde el repositorio del proyecto.
2. Extraiga el contenido del archivo en una carpeta de su elección (ejemplo: C:\RoutingApp).
3. Asegúrese de que la carpeta data esté presente junto al ejecutable oep.exe.
4. Haga doble clic en oep.exe para iniciar la aplicación.

#### **3.2. Linux**

##### Instalación desde Ejecutable Precompilado

1. Descargue el archivo .tar.gz del release más reciente.
2. Extraiga el archivo: routing-app-linux.tar.gz
3. Asegúrese de que la carpeta data esté presente.

### **4. Descripción de la Interfaz**

La interfaz está diseñada en un formato de panel dividido, el panel izquierdo contiene todos los controles de configuración y entrada de datos, y el área principal derecha está dedicada a la visualización del mapa y los resultados.

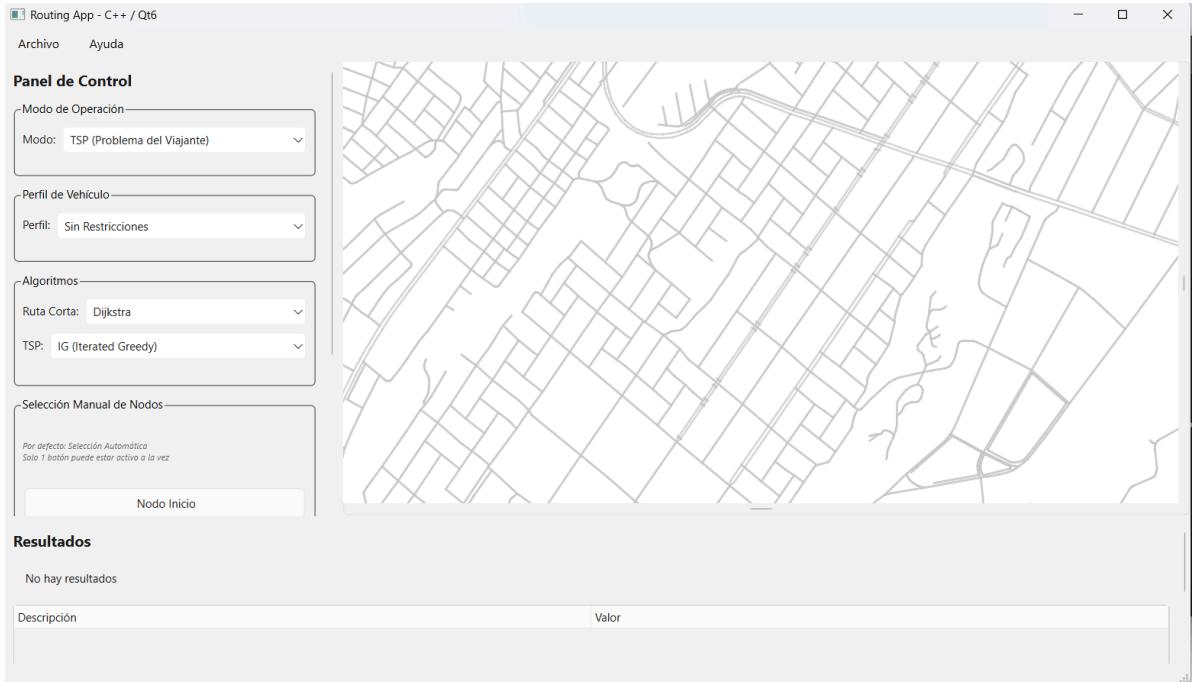


Figura 1: Interfaz principal de Routing App

## 4.1. Panel Izquierdo de Control

### 4.1.1. Configuración de Operación

**Modo de Operación:** Permite seleccionar el tipo de problema a resolver, se puede escoger entre TSP (Problema del Viajante) o Ruta corta (Pathfinding).

**Perfil de Vehículo:** Define las restricciones o características de la ruta, se puede elegir entre el modo sin restricciones, automóvil o peatón.

### 4.1.2. Algoritmos

Esta sección permite elegir qué algoritmos se utilizarán para resolver los problemas de enrutamiento seleccionados.

**Ruta Corta:** Algoritmo para encontrar el camino más corto entre dos puntos, se puede elegir entre Dijkstra, A\* y ALT.

**TSP:** Algoritmo de optimización para el problema del viajante, se puede elegir entre IG (Iterated Greedy), ILS\_B (Iterated Local Search B) e IGSA (IG + Simulated Annealing).

### 4.1.3. Selección Manual de Nodos

Esta sección permite al usuario definir manualmente los puntos de la ruta sobre el mapa.

**Nodo Inicio:** Botón para marcar el punto de partida de la ruta.

**Nodo/s Destino:** Botón para marcar los puntos intermedios o finales que el viajante debe visitar.

#### **4.1.4. Opciones TSP**

Esta es una casilla de verificación que define si la ruta calculada debe ser un circuito cerrado.

Si está marcada, el algoritmo debe asegurar que la ruta finalice en el mismo punto donde comenzó.

Si no está marcada, se resolvería un problema de "Path TSP", donde el objetivo es encontrar la ruta más corta que visite todos los puntos, sin necesidad de regresar al inicio.

#### **4.1.5. Acciones Principales**

Estos botones son las principales herramientas de control y ejecución de la aplicación.

**Limpiar Selección:** Permite borrar todos los nodos que hayan sido marcados en el mapa, restableciendo la entrada.

**Resolver TSP:** Una vez que se han configurado todos los parámetros, este botón inicia el cálculo y la ejecución del algoritmo de optimización seleccionado para encontrar la mejor ruta.

### **4.2. Panel Derecho: Área de Visualización**

Es el área principal y muestra una representación gráfica del mapa en escala de grises o blanco y negro, una representación pura del grafo de la red vial. Esta área sirve como lienzo para la interacción, permitiendo la selección de nodos y la visualización de la ruta o solución calculada.

### **4.3. Panel inferior: Resultados**

Esta es una sección dedicada a mostrar el output o los resultados de la ejecución del algoritmo. Presenta una tabla con dos columnas para mostrar la descripción y el valor de las métricas.

## 5. Gestión de Nodos

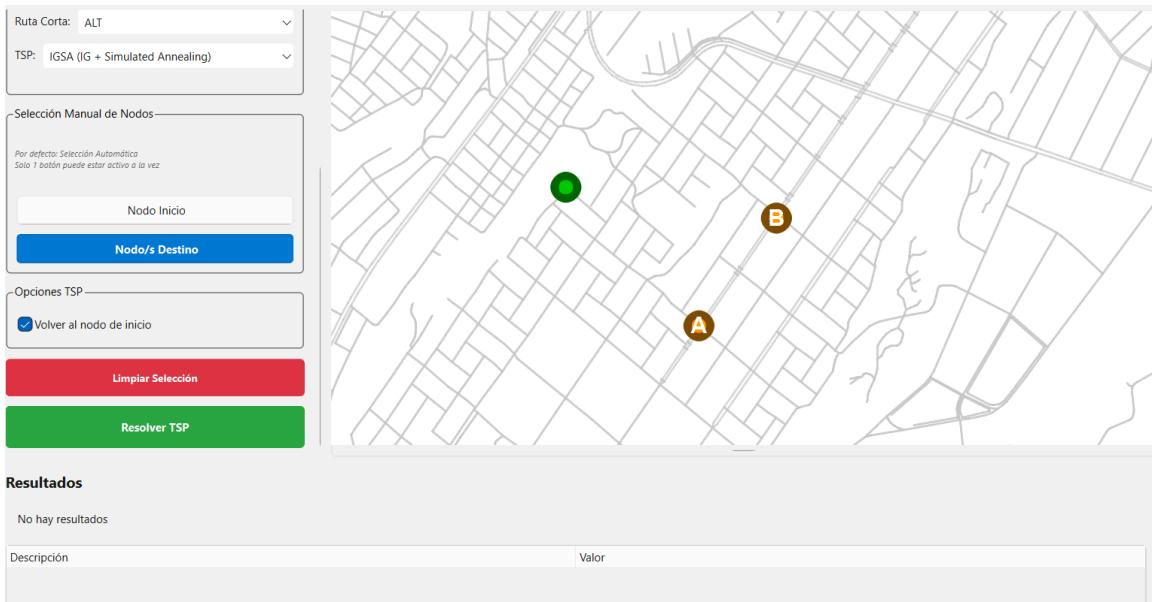


Figura 2: Ejemplo de selección de nodos en la interfaz

### 5.1. Modos de selección

La aplicación opera bajo un principio simple: solo un botón puede estar activo a la vez.

- Al hacer clic en un punto en el mapa, este queda marcado como un nodo seleccionado.
- Al activar un botón, cualquier otro modo de selección manual se desactiva automáticamente.

**Nota:** Por defecto, la aplicación puede estar configurada para la selección automática de nodos si no se presiona ninguno de estos botones.

### 5.2. Definición del nodo inicio

Al hacer clic en este botón, el usuario puede seleccionar un punto en el mapa. Este punto se representará como un círculo verde en el mapa y será el origen del recorrido.

### 5.3. Definición de los nodos destino

Al hacer clic en este botón, el usuario puede seleccionar múltiples puntos en el mapa. Estos puntos se representarán como círculos de color marrón, etiquetados con letras (A, B, C...) y serán los destinos que se deben incluir en la ruta.

### 5.4. Botones de Control

Estos botones inician o resetean el proceso de definición de nodos y cálculo de ruta.

- **Limpiar Selección:** Al presionar este botón se eliminarán todos los nodos que hayan sido marcados en el mapa, volviendo al inicio.

- **Resolver TSP:** Una vez que haya definido el nodo de inicio y al menos un nodo destino, al presionar este botón se ejecutará el algoritmo de optimización y se mostrará en pantalla la ruta más corta que cumpla con todos los requisitos.

## 6. Características Adicionales

### 6.1. Perfil de vehículo

Al seleccionar el perfil de un automóvil, en la representación gráfica se mostrarán algunas marcas rojas. Estas simbolizan las calles y rutas que no están permitidas para automóviles en la vida real, por lo que se bloquean esas calles como inaccesibles al momento de calcular la ruta más corta.



Figura 3: Ejemplo de marcación de rutas no accesibles

## 7. Casos de Uso

### 7.1. Caso 1: Planificación de una ruta cerrada (TSP Estándar)

**Objetivo:** planificar la ruta más corta que visite un conjunto de puntos y regrese al punto de partida.

1. Seleccionar el modo de operación: TSP (Problema del Viajante).
2. Seleccionar el algoritmo de TSP que se desea usar.
3. Hacer clic en el botón Nodo Inicio y luego hacer clic en el mapa para marcar el punto de partida.
4. Hacer clic en el botón Nodo/s Destino y hacer clic en el mapa para marcar al menos dos puntos intermedios a visitar.
5. Asegurar que la casilla de Volver al nodo de inicio esté marcada.
6. Presionar el botón “Resolver TSP”.

7. El sistema muestra la secuencia de la ruta más corta, la distancia y el tiempo total.

### **7.2. Caso 2: Planificación de una ruta abierta (Path TSP)**

**Objetivo:** planificar la ruta más corta que visite un conjunto de puntos, pero termine en el último destino sin regresar al origen.

1. Realizar los pasos 1 a 4 del Caso 1.
2. Desmarcar la casilla de Volver al nodo de inicio.
3. Presionar el botón “Resolver TSP”.
4. El sistema muestra la secuencia de la ruta más corta, la distancia y el tiempo total.

### **7.3. Caso 3: Cálculo de la ruta corta entre dos puntos**

**Objetivo:** encontrar la ruta más eficiente, corta o rápida, entre un punto de inicio y un único punto de destino.

1. Cambiar el modo de operación a Ruta Corta (Pathfinding).
2. Seleccionar el algoritmo de Ruta Corta que se desea usar.
3. Hacer clic en el botón Nodo Inicio y luego hacer clic en el mapa para marcar el punto de partida.
4. Hacer clic en el botón Nodo/s Destino y hacer clic en el mapa para marcar un único punto de llegada.
5. Presionar el botón “Calcular Ruta Corta”.
6. El sistema muestra gráficamente la ruta más eficiente.

### **7.4. Caso 4: Limpieza y reconfiguración de la entrada**

**Objetivo:** reiniciar los datos de entrada.

1. Presionar el botón “Limpiar Selección”
2. Todos los nodos desaparecen del mapa.
3. Procede a configurar un nuevo modo de operación y seleccionar nuevos nodos.

## **8. Preguntas Frecuentes**

### **8.1. ¿Cuántos nodos puedo seleccionar para resolver el TSP?**

La aplicación está optimizada para resolver TSP con 3-20 waypoints. Con más de 20 puntos, el tiempo de cálculo puede aumentar significativamente (varios minutos). El límite práctico depende del hardware disponible.

### **8.2. ¿Qué diferencia hay entre los algoritmos de ruta corta (Dijkstra, A, ALT)?\***

**Dijkstra:** Algoritmo clásico, garantiza la ruta óptima, explora más nodos.

**A:\*** Utiliza heurística de distancia euclídea, más rápido que Dijkstra en la mayoría de casos.

**ALT:** Optimización avanzada con landmarks, ideal para grafos grandes (rendimiento superior).

En la práctica, A\* es la mejor opción para uso general.

### **8.3. ¿Puedo usar la aplicación sin conexión a internet?**

Sí, la aplicación funciona completamente offline. Todos los datos del mapa se cargan desde archivos binarios locales en la carpeta data/graphs/.

### **8.4. ¿Cómo afecta el perfil de vehículo a la ruta calculada?**

**Sin restricciones:** Usa todas las calles disponibles sin limitaciones.

**Automóvil:** Bloquea calles peatonales y rutas con restricción vehicular (marcadas en rojo).

**Peatón:** Permite el uso de todas las calles, incluyendo zonas peatonales.

### **8.5. ¿Por qué algunos algoritmos TSP tardan más que otros?**

Los algoritmos tienen diferentes complejidades:

**IG (Iterated Greedy):** Rápido, buenas soluciones en pocos segundos.

**ILS\_B (Iterated Local Search):** Más lento, realiza búsquedas locales intensivas.

**IGSA (IG + Simulated Annealing):** El más lento, busca soluciones de máxima calidad mediante exploración estocástica.

## 9. Solución de Problemas

### 9.1. La aplicación no inicia

Al hacer doble clic en el ejecutable, no aparece ninguna ventana.

Soluciones:

1. Verificar dependencias de Qt (Windows):
  - Asegúrese de que las DLLs de Qt estén en la misma carpeta que el ejecutable o en el PATH del sistema.
  - DLLs necesarias: Qt6Core.dll, Qt6Gui.dll, Qt6Widgets.dll (para Qt6).
  - Si compiló con Qt5, busque las versiones equivalentes (Qt5Core.dll, etc.).
2. Comprobar biblioteca Visual C++ Redistributable (Windows):
  - Descargue e instale el paquete desde: Microsoft Visual C++ Redistributable
3. Verificar permisos de ejecución (Linux/macOS).

### 9.2. No aparece el mapa

La aplicación inicia pero el área del mapa está en blanco o gris.

Soluciones:

1. Verificar la carpeta data:
  - Asegúrese de que existe la carpeta data/graphs/ en el mismo directorio que el ejecutable.
  - Confirme que el archivo arequipa.bin está presente y no está corrupto.
  - Tamaño esperado del archivo: aproximadamente 5-15 MB.
2. Permisos de lectura:
  - Verifique que la aplicación tiene permisos para leer el archivo binario.

### 9.3. Error al ejecutar algoritmos

Al presionar "Resolver TSP" o "Calcular Ruta Corta", aparece un mensaje de error o la aplicación se congela.

Soluciones:

1. Nodos inválidos seleccionados:

- Asegúrese de hacer clic directamente sobre intersecciones o calles válidas (nodos del grafo).
- Si clieca en áreas vacías, es posible que no se seleccione ningún nodo.
- Use la función "Limpiar Selección" y vuelva a marcar los puntos.

2. Número insuficiente de waypoints (TSP):

- Para resolver TSP, debe seleccionar al menos 3 destinos además del nodo de inicio.
- Error típico: "Se requieren al menos 3 waypoints para resolver TSP".

3. Falta de conexión entre nodos:

- Si dos puntos están en componentes desconexas del grafo (sin camino entre ellos), el algoritmo puede fallar.
- Verifique que los nodos estén conectados por calles en el mapa.
- Pruebe con el perfil "Sin restricciones" para aumentar las opciones de ruta.

4. Consumo excesivo de memoria:

- Con muchos waypoints (>20), el cálculo de la matriz TSP puede consumir mucha RAM.
- Solución: Reduzca el número de destinos o cierre otras aplicaciones.

5. Aplicación congelada durante cálculos largos:

- Los algoritmos TSP pueden tardar varios segundos o minutos con muchos waypoints.
- No cierre la aplicación, espere a que termine el cálculo.
- En futuras versiones se implementará una barra de progreso más detallada.

6. Crash durante ejecución:

- Si la aplicación se cierra inesperadamente:
  - Actualice a la versión más reciente del software.
  - Reporte el problema en el repositorio con los pasos exactos para reproducirlo.
  - Ejecutar desde terminal para capturar el mensaje de error completo.

## 10. Glosario

**Grafo** Estructura de datos con nodos y aristas.

**Nodo** Vértice del grafo (lugar turístico).

**Dijkstra** Algoritmo para encontrar la ruta mínima.

**A\*** Extensión heurística de Dijkstra.

**ALT** Algoritmo para encontrar la ruta mínima.

**IG** Algoritmo metaheurístico para resolver problemas de optimización complejos.

**ILS\_B** Metaheurística que alterna entre búsqueda local intensiva y perturbaciones estratégicas.

**IGSA** Algoritmo de optimización híbrido para encontrar mejores soluciones con mayor rapidez.

## 11. Contacto y Soporte

Para soporte técnico o consultas:

**GitHub:** <https://github.com/rescobedoq/oep>

**Email:** [rminacal@unsa.edu.pe](mailto:rminacal@unsa.edu.pe)