

# Sistema Nacional de Tipificación de Emergencias Potenciales (SiNTiEmPo)

Camila Rodríguez Águila, C36624, camila.rodriguezagUILA@ucr.ac.cr

**Resumen**—El Sistema Nacional de Tipificación de Emergencias Potenciales (SiNTiEmPo) gestiona una red de centros de distribución ubicados en distintas ciudades del país, las cuales están interconectadas mediante rutas con tiempos promedio de traslado conocidos. Este proyecto representa dicha red como un grafo dirigido y ponderado, e implementa un sistema capaz de responder consultas estratégicas clave, como determinar la mejor ubicación para almacenar suministros, identificar las rutas más cortas entre centros, encontrar el par de ciudades con mayor o menor distancia, y clasificar las ciudades según su eficiencia de conexión. Para calcular las distancias mínimas entre todos los pares de ciudades, se utiliza el algoritmo de Floyd-Warshall, elegido por su eficacia y adecuación en grafos densos con pesos positivos. El sistema permite la carga de datos a través de archivos CSV y ofrece una interfaz interactiva para realizar consultas sobre la red logística.

**Palabras clave**—Floyd-Warshall, Algoritmos, Rutas, Corto, Rápido

## I. INTRODUCCIÓN

El Sistema Nacional de Tipificación de Emergencias Potenciales (SiNTiEmPo) opera una red compuesta por centros de distribución ubicados en diversas ciudades del país. Estos centros almacenan recursos esenciales para la atención de emergencias. Algunas ciudades están unidas por rutas directas, y para cada una se dispone del tiempo estimado de traslado.

Este proyecto propone una representación de la red mediante un grafo dirigido con pesos, en el que cada nodo simboliza una ciudad y cada arista indica una conexión directa, ponderada por el tiempo promedio de viaje. Con base en este modelo, se construye un sistema que facilita la resolución de consultas clave para la gestión y respuesta ante emergencias. Entre las funcionalidades destacan: el análisis para ubicar de forma óptima los centros de almacenamiento, la búsqueda de rutas mínimas entre pares de ciudades, las ciudades más cercanas y más lejanas entre sí, y la clasificación de las ciudades según su grado de eficiencia en la red.

Para llevar a cabo estos análisis, se implementa el algoritmo de Floyd-Warshall, el cual resulta adecuado por su eficiencia en grafos densamente conectados y su capacidad para hallar los caminos más cortos entre todos los pares de nodos. De esta forma, el sistema permite importar la información de la red desde archivos CSV y ofrece una interfaz amigable e interactiva para el acceso a las funcionalidades disponibles.

## II. METODOLOGÍA

### II-A. Representación del Grafo

El sistema representa el grafo mediante una lista de adyacencia implementada con vectores de C++. Cada ciudad

se almacena como un vértice con un identificador único y un nombre, mientras que las conexiones se representan como aristas dirigidas con pesos que indican el tiempo de viaje.

La clase `Grafo` maneja la estructura de datos principal, incluyendo:

- Un vector de vértices que contiene información de cada ciudad
- Un mapa hash para realizar búsquedas eficientes por nombre de ciudad
- Métodos para agregar vértices y aristas dinámicamente

### II-B. Algoritmo de Floyd-Warshall

Para calcular las distancias mínimas entre todos los pares de ciudades, se emplea el algoritmo de Floyd-Warshall. Este algoritmo fue seleccionado por las siguientes razones:

- Calcula todas las distancias mínimas en una sola ejecución con complejidad  $O(V^3)$
- Es apropiado para grafos densos donde se requieren múltiples consultas
- Permite la reconstrucción de caminos mediante una matriz de predecesores
- Maneja eficientemente grafos con pesos positivos

El algoritmo se implementa en el método `floydWarshall` de la clase `Grafo`, generando dos matrices:

- Matriz de distancias mínimas entre todos los pares de vértices
- Matriz de siguiente vértice para reconstruir caminos óptimos

### II-C. Análisis de Consultas

La clase `AnalizadorGrafo` implementa las cinco consultas principales:

**1. Centro de distribución óptimo:** Se calcula la suma de distancias mínimas desde cada ciudad hacia todas las demás. La ciudad con la suma menor es la más eficiente para centralizar la distribución.

**2. Mejor dispatcher:** Para una ciudad destino específica, se identifican todas las ciudades origen y se selecciona la que tenga la menor distancia mínima hacia el destino.

**3. Ciudades más distantes:** Se recorren todos los pares de ciudades en la matriz de distancias mínimas para encontrar el par con la mayor distancia.

**4. Ciudades menos distantes:** Similar al anterior, pero buscando el par con la menor distancia (excluyendo la distancia cero de una ciudad hacia sí misma).

**5. Ranking por tiempo promedio:** Se calcula el promedio de distancias mínimas desde cada ciudad hacia todas las demás, ordenándolas de menor a mayor tiempo promedio.

#### II-D. Carga de Datos

El sistema incluye la clase `FileReader` que procesa archivos CSV con el formato:

- ID de ciudad origen
- Nombre de ciudad origen
- ID de ciudad destino
- Nombre de ciudad destino
- Tiempo de viaje

El parser maneja automáticamente la eliminación de espacios en blanco y la creación dinámica de vértices según aparecen en el archivo.

### III. RESULTADOS

El sistema fue evaluado con tres conjuntos de datos de diferente escala (Large, Medium y Small), demostrando la eficacia del algoritmo de Floyd-Warshall para resolver consultas en redes de ciudades con distintas topologías.

#### III-A. Organización de Resultados

Los resultados se organizan en dos categorías principales:

- **Resúmenes:** Documentos PDF que contienen un análisis conciso de cada conjunto de datos:
  - Large.pdf
  - Medium.pdf
  - Small.pdf
- **Salidas:** Contiene:
  - Archivos con la salida textual directa de la consola para cada caso (Large, Medium, Small)
  - Anexo General (en formato DOCX y PDF): Documento consolidado con todas las salidas organizadas y un índice para facilitar la navegación

#### III-B. Análisis de Rendimiento

El sistema demostró las siguientes características de complejidad temporal:

- Carga de datos:  $O(E)$  (donde  $E$  es el número de aristas)
- Ejecución de Floyd-Warshall:  $O(V^3)$  (donde  $V$  es el número de vértices)
- Consultas individuales:  $O(V^2)$  en el peor caso

El sistema mantuvo tiempos de respuesta interactivos incluso para redes de varios cientos de ciudades, validando su utilidad para aplicaciones prácticas.

### IV. MANUAL DE COMPILACIÓN

Para compilar el programa, siga los siguientes pasos:

1. Clonar el repositorio del proyecto.
2. Ir al directorio del proyecto `Tareas_Programadas/tp3`
3. Ejecutar el comando `make` en la terminal para compilar el programa.

4. Para ejecutar el programa, utilizar el comando `./sintiempo <archivo.csv>` donde `<archivo.csv>` es el archivo de datos a procesar (por ejemplo: `tests/input_large.csv`).  
Ejemplo: `Tareas_Programadas/tp3$ ./bin/sintiempo`
1. Clonar el repositorio del proyecto
2. Navegar al directorio del proyecto:  
`cd Tareas_Programadas/tp3`
3. Compilar el programa:  
`make`
4. Ejecutar el programa con uno de los archivos de prueba:  
`./bin/sintiempo tests/input_small.csv`  
o para el archivo grande:  
`./bin/sintiempo tests/input_large.csv`

### V. MANUAL DE USO

Al iniciar el programa, se presenta un menú principal con las siguientes opciones:

1. **Centro de distribución más efectivo:** Calcula la ciudad óptima para centralizar la distribución de suministros.
2. **Mejor ciudad para despacho:** Sigue el nombre de una ciudad destino e identifica la mejor ciudad origen para el despacho.
3. **Par de ciudades más distantes:** Encuentra el par de ciudades con mayor tiempo de viaje entre ellas.
4. **Par de ciudades menos distantes:** Encuentra el par de ciudades con menor tiempo de viaje entre ellas.
5. **Ranking de ciudades por tiempo promedio:** Muestra todas las ciudades ordenadas por su tiempo promedio de viaje.
6. **Información del grafo cargado:** Presenta estadísticas básicas del grafo procesado.
7. **Salir:** Termina la ejecución del programa.

Para su uso, digite el número de la pregunta que desea consultar y siga las instrucciones presentadas en la consola.

### VI. CONCLUSIONES

El sistema SiNTiEmPo desarrollado proporciona una solución eficiente para el análisis de redes de distribución de emergencias. La implementación del algoritmo de Floyd-Warshall permite responder múltiples consultas de manera rápida y precisa.