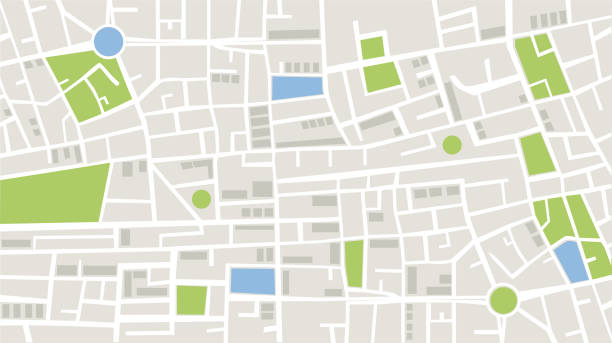
Programación II TPO “Logística Urbanística mediante grafos”

Nombre: Clerc,Sebastian Yanick.

Materia:Programación II.

Fecha de entrega:27/06/25.

Profesor: Perez,Nicolas Ignacio .



UADE 2025

### 1. Introducción del proyecto

En este proyecto decidimos abordar un problema real que pudiera resolverse utilizando algoritmos de grafos ponderados. En lugar de trabajar con un escenario complejo como una red aeroportuaria o logística global, optamos por representar una red urbana simplificada, compuesta por ubicaciones conocidas como una plaza, una escuela, un hospital, un supermercado y una estación de tren.

El objetivo fue simular un sistema que permitiera calcular las rutas más convenientes entre estos puntos, considerando como peso la distancia entre ellos. Para resolver este problema, implementamos el algoritmo de Dijkstra, el cual nos permite encontrar el camino más corto desde un punto de origen hacia todos los demás puntos del grafo.

A través del uso de estructuras modulares siguiendo el paradigma de Tipo de Dato Abstracto (TDA), desarrollamos clases como Vértice, Arista y Grafo, que nos ayudaron a organizar mejor el código, facilitar su comprensión y dejarlo preparado para futuras ampliaciones.

### 2. Problemática del proyecto

Nos planteamos una problemática concreta: ¿cómo podríamos calcular la ruta más corta entre dos puntos dentro de una pequeña red urbana?. Imaginamos un sistema de navegación GPS a escala reducida, que nos permitiera simular la elección de rutas dentro de una ciudad, barrio o campus.

Este tipo de análisis es útil en múltiples escenarios:

* Planificación de rutas escolares.
* Trazado de caminos más rápidos para ambulancias o patrullas.
* Optimización de entregas a domicilio.
* Rutas peatonales seguras y cortas.

Entendimos que modelar este tipo de sistema requería una estructura que representará lugares y conexiones, junto con un algoritmo que determinara eficientemente cuál era el camino más conveniente.

### 3. Solución de la problemática

La solución que propusimos consistió en representar la red urbana como un grafo no dirigido y ponderado, donde:

* Cada vértice representa un lugar (como una escuela o una plaza).
* Cada arista representa una calle o conexión entre dos lugares.
* El peso de la arista indica la distancia entre esos dos puntos (en kilómetros).

Diseñamos tres TDAs para organizar mejor la solución:

* Vértice: contiene el nombre del lugar.
* Arista: almacena el destino y el peso (distancia).
* Grafo: se encarga de manejar las conexiones entre los vértices y ejecutar el algoritmo de Dijkstra.

Con esta estructura, implementamos el algoritmo de Dijkstra de forma clara y sencilla. Inicialmente asignamos a todos los vértices una distancia infinita, excepto al nodo de inicio, que recibe valor 0. Luego recorrimos todos los vecinos de cada nodo, actualizando sus distancias si encontrábamos una ruta más corta.

Esta solución nos permitió encontrar las rutas mínimas desde un punto de partida hacia el resto del grafo, facilitando la toma de decisiones logísticas en función del costo (en este caso, distancia).

### 4. Objetivo del proyecto

El objetivo general de nuestro proyecto fue modelar una red de rutas urbanas utilizando grafos y aplicar sobre ella el algoritmo de Dijkstra para determinar rutas óptimas desde un nodo origen.

Entre los objetivos específicos que nos planteamos, destacamos los siguientes:

* Representar de forma estructurada una red urbana con vértices y aristas ponderadas.
* Implementar el algoritmo de Dijkstra con una lógica sencilla, orientada a la comprensión y la claridad.
* Evaluar los resultados obtenidos desde distintos puntos de inicio.
* Comparar nuestra versión del algoritmo con la versión vista en clase, más optimizada.

Además, buscábamos que el sistema fuera fácil de ampliar, por ejemplo, permitiendo agregar nuevos puntos al grafo o incorporar más atributos (como tiempo de viaje o costo económico).

### 5. Algoritmos implementados

#### 5.1 Dijkstra

Decidimos implementar el algoritmo de Dijkstra porque es ideal para hallar caminos mínimos en grafos con pesos no negativos. En nuestro caso, como los pesos representan distancias (en kilómetros), no existen valores negativos.

La implementación que hicimos parte de una lista de adyacencia y una búsqueda lineal para seleccionar el nodo con menor distancia aún no visitado. Aunque sabíamos que usar una cola de prioridad (como en la versión vista en clase) hubiese sido más eficiente, elegimos este enfoque para mantener la claridad del código, ya que nuestro grafo tiene pocos nodos.

Durante la ejecución, el algoritmo va actualizando las distancias mínimas conocidas desde el nodo origen a cada vértice. Al finalizar, tenemos una tabla de distancias óptimas desde el punto de partida hacia el resto de los nodos del grafo.

### 7. Ventajas del sistema

Nuestro sistema tiene varias ventajas importantes:

* Es intuitivo y fácil de entender, ideal para estudiantes que estén comenzando a trabajar con grafos.
* Usa el paradigma TDA, lo cual nos permite modificar o ampliar partes del programa sin romper el resto.
* Puede ser escalado fácilmente con nuevos nodos o funcionalidades adicionales (como tiempos de viaje, semáforos, etc.).
* Permite evaluar diferentes escenarios logísticos con muy poca configuración.

### 8. Conclusión

En este trabajo pudimos aplicar los conocimientos de estructuras de datos y algoritmos aprendidos durante la cursada, desarrollando una solución concreta a una problemática real y cotidiana.

El uso de grafos ponderados y del algoritmo de Dijkstra nos permitió modelar rutas urbanas de forma eficiente y clara. A su vez, trabajar con TDAs nos ayudó a pensar en un diseño modular, escalable y fácil de mantener.

Comparar nuestra implementación con la versión optimizada de clase también nos permitió entender que existen distintos niveles de complejidad para un mismo problema, y que no siempre es necesario comenzar por la solución más eficiente si lo que buscamos es aprender los fundamentos.

En definitiva, el sistema que desarrollamos es una buena base para futuras mejoras, y podría ser adaptado a situaciones más reales agregando visualizaciones, entradas dinámicas o distintos tipos de peso (tiempo, costo, emisiones, etc.). Nos llevamos una experiencia muy positiva, tanto a nivel técnico como conceptual.