



# Proyecto Final

## Estructuras de Datos y Análisis de Algoritmos

# Integrantes

Ana María Fernández Zapata – 2240059

Alejandra Díaz Navarro – 2240063

Cristian David Quintana – 2240072

# Descripción del Problema 1

En la ciudad de Bucaramanga, el sistema de transporte público Metrolínea busca implementar una solución tecnológica para la gestión dinámica de rutas y paradas de sus buses. La idea principal es crear un sistema en el que:

- Cada ruta del Metrolínea esté compuesta por una secuencia de paradas ordenadas.
- Se puedan crear nuevas rutas, eliminar rutas existentes, y modificar las paradas de cada ruta de manera sencilla.
- A medida que el bus avanza en su recorrido, la parada actual se elimine automáticamente de la lista, dejando únicamente las paradas restantes.
- Al finalizar el recorrido, la lista de paradas de esa ruta queda vacía o se reinicia automáticamente para el siguiente recorrido.

Este sistema busca optimizar el seguimiento y la visualización del progreso del bus en tiempo real, permitiendo también la modificación rápida de rutas por causas como cierres viales, desvíos o nuevas zonas de cobertura.

# Primera Entrega: Listas Enlazadas

Para la primera entrega, se implementó un modelo de listas enlazadas que representaran una ruta para cualquier bus de Metrolínea, de esta forma, las funcionalidades que implementamos fueron:

- Cada parada del Metrolínea es modelada como un nodo en una lista enlazada simple. Esto permite que las rutas puedan cambiar en tiempo real (añadir, eliminar paradas) sin afectar el resto del sistema.
- Las paradas se recorren en **orden**, como ocurre naturalmente en un recorrido de bus.
- Modelo basado en pila (último en entrar, primero en salir)
- El método agregar inserta siempre al inicio de la lista. Por lo tanto, la última parada agregada será la primera en mostrarse (no refleja un orden de recorrido realista, pero es útil para demostrar la estructura).
- Eliminación automática de paradas al avanzar el bus:
- Cada vez que se llama eliminarParada(), se elimina la primera parada de la lista (la cabeza), simulando que el bus ya pasó por esa estación.

# Primera Entrega: Ventajas y Desventajas

## Ventajas

- Es fácil agregar o quitar paradas sin necesidad de desplazar elementos (como ocurriría en un arreglo).
- Se utiliza solo la memoria necesaria para el número actual de paradas, sin desperdicio.
- Ideal para estructuras secuenciales que cambian constantemente, como el recorrido de un bus que avanza y modifica su lista de paradas en tiempo real.
- Se pueden crear rutas desde cero, modificar paradas intermedias o reordenarlas fácilmente.

## Desventajas

- No se puede acceder directamente a una parada específica sin recorrer la lista desde el inicio.
- Cada nodo necesita almacenar referencias (punteros), lo que aumenta el consumo de memoria frente a estructuras como arrays.
- Operaciones como buscar una parada específica o realizar modificaciones condicionales pueden ser más lentas si no se optimizan.
- Los errores en los enlaces entre nodos (como enlaces perdidos o bucles) pueden ser difíciles de detectar.

# Descripción del Problema 2

Un grupo de colombianos abrió un restaurante de tamales llamado "**La Ruta del Tamal**", ubicado en la Plaza Moreno de la ciudad de La Plata, Argentina. Gracias a la autenticidad de su sabor y la gran acogida por parte del público, el restaurante comenzó a recibir una creciente cantidad de pedidos a domicilio. Esto generó la necesidad de implementar un sistema eficiente de reparto de pedidos en toda la ciudad.

Inicialmente, los encargados del restaurante se vieron en la necesidad de modelar las entregas de manera básica, considerando cada punto de entrega como un nodo y las conexiones entre ellos como rutas. Sin embargo, con el crecimiento de la demanda, esta estructura básica resultó insuficiente, lo que llevó a replantear el sistema para hacerlo más flexible y eficiente, permitiendo rutas alternativas, ponderación de distancias en tiempo, y optimización de recorridos.

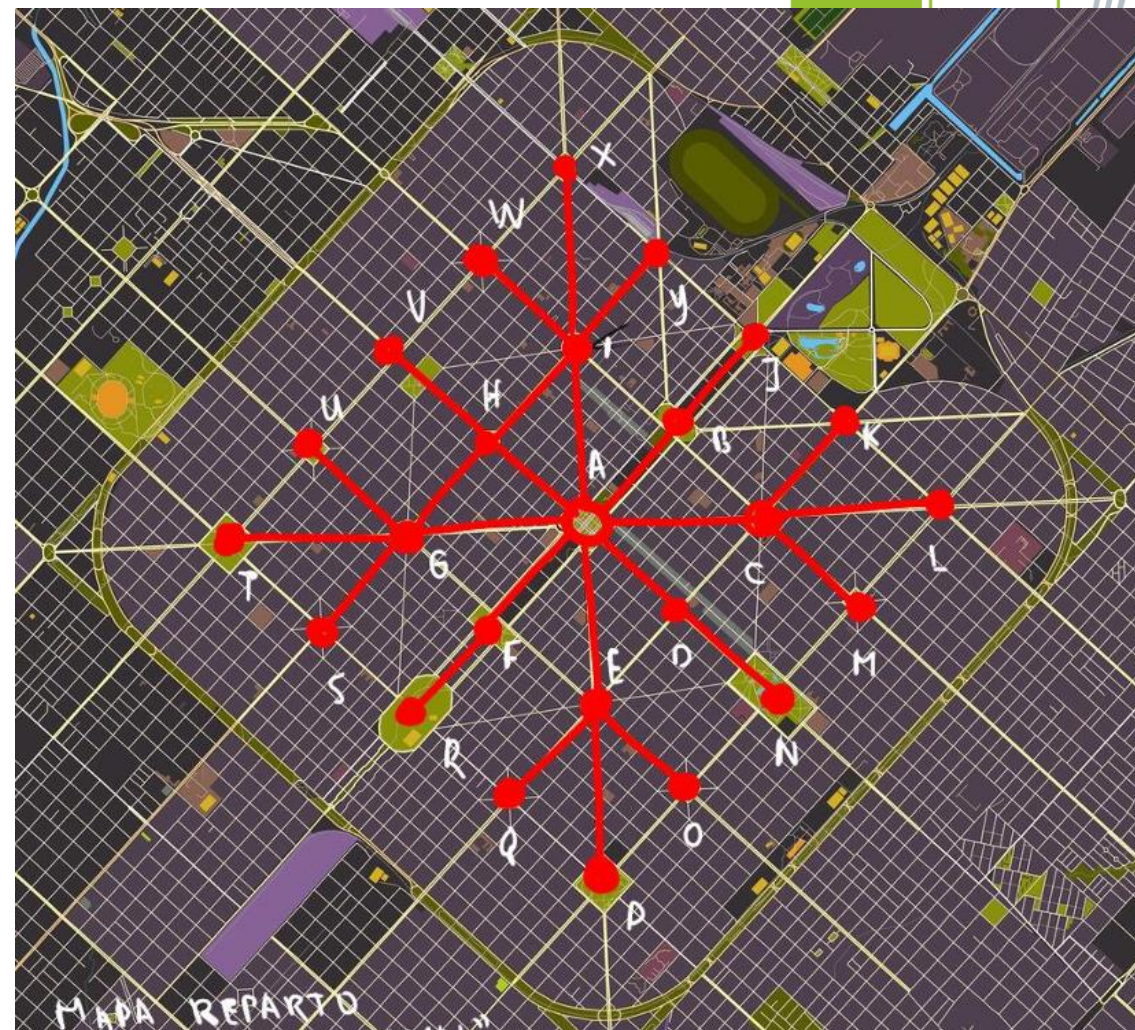


## Segunda Entrega: Sistema de Árbol

En la segunda entrega del proyecto, se implementó un modelo de entregas utilizando un árbol, donde:

- Cada nodo representaba un domicilio o punto de entrega.
- El nodo raíz representaba el restaurante (ubicado en Plaza Moreno).
- Las conexiones entre nodos eran unidireccionales y seguían una estructura jerárquica (padre-hijo), sin ciclos ni rutas alternativas.

Además, se realizó un menú para que los clientes pudieran realizar pedidos.



**Imagen 1:** Primer Mapa de Reparto para La Ruta del Tamal

# Segunda Entrega: Ventajas y Desventajas

## Ventajas

- Permite representar jerarquías con claridad.
- Solo hay un camino único entre el restaurante (raíz) y cada domicilio, lo cual facilita los algoritmos de recorrido (DFS, BFS).
- Ideal para una primera aproximación al problema logístico.

## Desventajas

- No se pueden modelar múltiples caminos entre domicilios.
- No permite volver a un nodo una vez que se ha recorrido.
- No hay forma de asignar pesos o tiempos de recorrido a las conexiones.
- No refleja la complejidad de una ciudad ni permite optimizar las entregas.



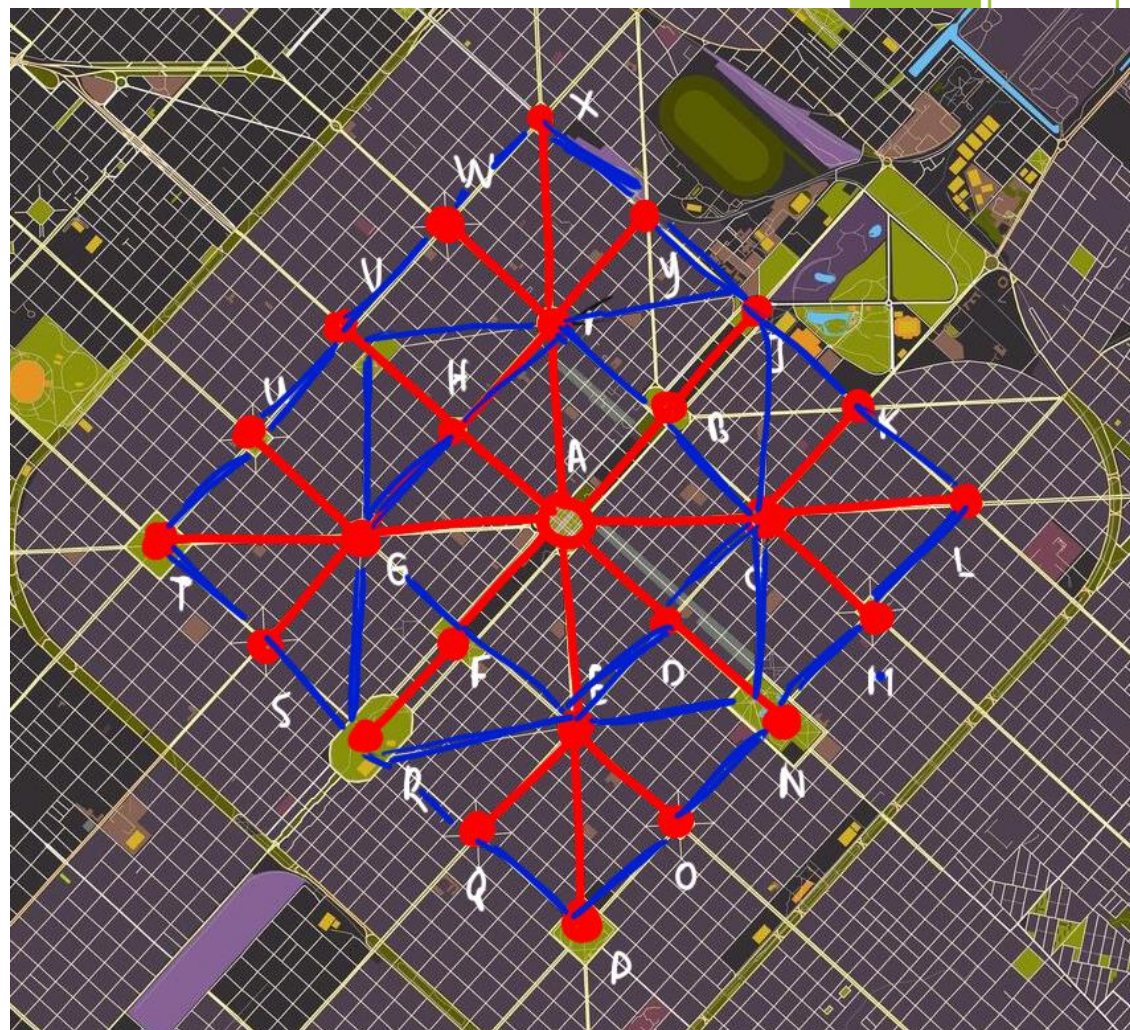
## Tercera Entrega: Sistema de Grafos

En la tercera y última entrega del proyecto, se rediseñó el sistema utilizando un grafo ponderado y no dirigido, lo cual permitió una mayor flexibilidad y realismo en la representación del mapa de entregas.

En este modelo:

- Cada nodo siguió representando un domicilio o punto de entrega.
- Las aristas representaban los caminos entre ubicaciones, y cada una fue ponderada con el tiempo estimado de recorrido (en minutos).

Al no tener una estructura jerárquica fija, se permitió conectar nodos que no seguían una relación padre-hijo, facilitando así la existencia de múltiples rutas entre puntos. Con este nuevo enfoque, fue posible implementar algoritmos de búsqueda y optimización de rutas.



**Imagen 2:** Segundo Mapa de Reparto para La Ruta del Tamal. Las Aristas Azules Representan las Nuevas Conexiones

# Tercera Entrega: Ventajas y Desventajas

## Ventajas

- Puede modelar múltiples caminos, intersecciones y ciclos.
- Permite asignar valores de tiempo o distancia a cada camino.
- Se pueden aplicar algoritmos como Dijkstra para encontrar la ruta más rápida.
- Se pueden agregar o eliminar conexiones fácilmente sin romper la estructura.

## Desventajas

- Requiere estructuras de datos más avanzadas y algoritmos más complejos.
- Necesita más planificación para evitar errores como ciclos infinitos o grafos no conexos.
- El comportamiento del sistema puede no ser tan predecible como en un árbol simple.

# ¿Cuál versión es mejor?

Claramente, el sistema de **Grafos** es mejor en este caso.

## ¿Por qué?

- Las ciudades como La Plata o Bucaramanga no tienen una estructura jerárquica como un árbol. Un árbol solo permite una única ruta desde el restaurante hacia cada domicilio. Esto no refleja la realidad urbana, donde existen varias rutas posibles.
- No se pueden asignar tiempos de recorrido entre puntos. Por lo tanto, no se puede elegir la opción más rápida.
- No es posible volver atrás ni tomar caminos alternativos, lo que impide redirigir una entrega si hay un problema (como tráfico o calles cerradas).
- Los grafos permiten modelar todos estos caminos y conexiones, cosa que un árbol o lista (estructura sin ciclos ni rutas alternativas) no puede hacer.





Universidad  
Industrial de  
Santander



¡Gracias!