

**Arquitectura de Software: Estimación de Velocidad Promedio por Tramos
en Rutas con Datos del SITM-MIO – Proyecto Final**

El Centro de Control de Operación (CCO) de Metrocali es la unidad responsable de la operación diaria del Sistema Integrado de Transporte Masivo de Occidente (SITM-MIO). Esta operación se debe ejecutar dando cumplimiento al Plan de Servicios de Operación (PSO) por parte de los concesionarios de la operación en la ciudad, así como de la gestión, control y seguimiento de sus flotas de buses respectivas (alrededor de 1000 buses en total, con proyección de crecimiento a 2.500). Diariamente, el sistema transporta a alrededor de 450.000 pasajeros, a través de 100 rutas principales. Cada bus está equipado con alrededor de 40 sensores conectados a un computador embebido, con un equipo de comunicaciones GPRS, que transmite los valores sensados aproximadamente cada 30 segundos a un DataCenter. El conjunto de datos que contiene los valores sensados se pone en un solo registro, el cual denomina **datagrama**.

Más concretamente, en la operación normal del sistema de transporte, los buses en operación envían eventos que incluyen desde los datos normales sensados, tales como la simple actualización de la posición geolocalizada por GPS, hasta la apertura y cierre de puertas, y otros adversos como ingreso no autorizado de pasajeros, atracos, avería de motor, trancón en la vía, o choque del bus con otro vehículo (algunos de estos eventos los envía el conductor mediante una interfaz de usuario especial en el computador embebido del bus). La cantidad de eventualidades que ocurren en el tráfico de la ciudad, y que deben enfrentar y resolver los controladores en la central de operación es abrumadora: el CCO puede recibir entre 2,500,000 y 3,000,000 de eventos por día en total, y actualmente no cuentan con herramientas eficaces para procesarlas adecuadamente.

Después de años de operación, se quiere explotar los datos que se han recogido. La organización quiere iniciar con un piloto con los datos de un año estos eventos, para brindar un servicio a la comunidad que indique la velocidad promedio de desplazamiento que se tiene cuando se quiere viajar entre dos puntos de la ciudad.

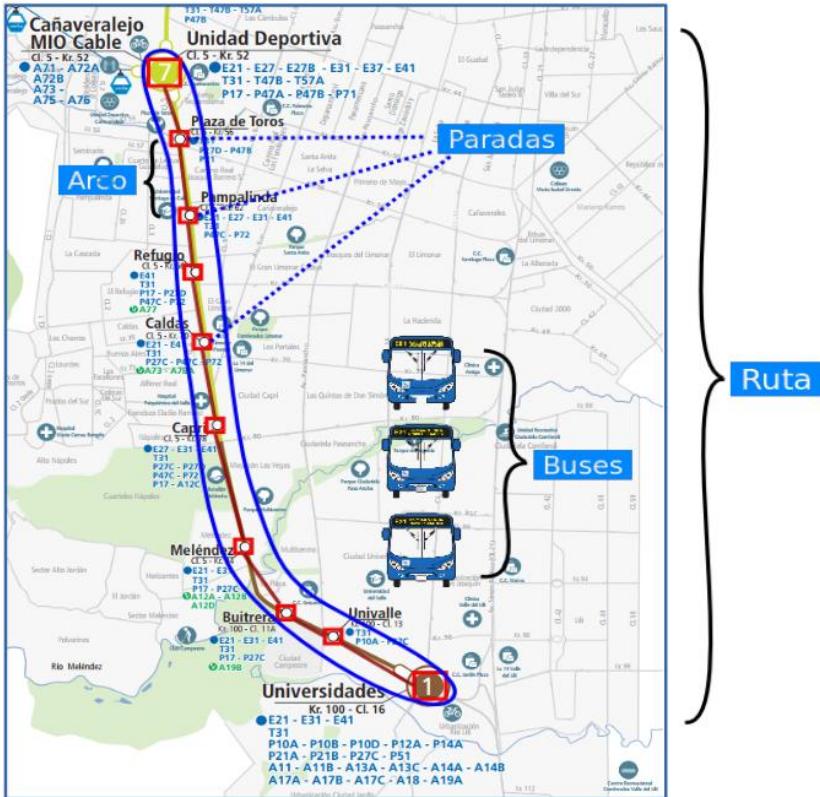


Figura 1. Rutas, Paradas, Arcos.

Requerimientos elicitudes para clasificar y analizar (F1: agrupar, jerarquizar, ordenar; F2: particionar, asignar, subespecificar) con los métodos que correspondan:

1	Los buses y estaciones generan eventos que pueden afectar la operación del SITM-MIO y deben ser tenidos en cuenta para crear eventos automáticamente, y según su categoría, asignarles una prioridad. El conductor debe poder enviar eventos (previamente parametrizados con su categoría y prioridad en el sistema) mediante la GUI que debe tener el computador del bus (ejemplo: menú → escoja el evento que quiere enviar: pinchazo de llantas, avería grave de motor, bloqueo o trancón en la vía, choque del bus).
2	Para evitar accidentes, el sistema debe permitir escoger los eventos a enviar mediante una perilla que al girarla pase el cursor por las opciones, y al presionarla, seleccione la opción sobre la que esté el cursor.
3	El sistema debe permitir administrar los roles, los usuarios y los permisos.
4	El sistema debe permitir visualizar el dónde están todos los buses de la operación del SITM-MIO en tiempo real en el mapa de Cali.
5	El diseño del sistema debe hacerse con diagramas en UML.
6	El sistema debe ser capaz de recibir y procesar los grandes volúmenes de datos generados por los buses del SITM-MIO, tales como las posiciones GPS de los buses, eventos operativos y reportes de los controladores, combinando información histórica y los datos que se reciben en tiempo real. Todos estos datos deben ser persistidos en base de datos.
7	El sistema debe realizar análisis que permitan estimar variables de interés relacionadas con la movilidad, tales como tiempos promedio de viaje por arco con base en datos históricos. Esta información debe poder actualizarse en tiempo real, considerando los datos históricos y los datos que vayan apareciendo en tiempo real.
8	El sistema debe permitir asignar rutas y zonas de la ciudad para ser supervisadas a un controlador de operación.

9	El sistema debe permitirle a cada uno de los 40 controladores de operación visualizar en tiempo real sus zonas asignadas, y para cada zona, la velocidad promedio por arco, de los arcos que estén en dicha zona.
10	El sistema debe tener una vista que le permita a los usuarios (controladores y administrador) iniciar y cerrar sesión.
11	El sistema debe mantener un análisis actualizado del comportamiento del SITM-MIO, adaptándose al crecimiento en el volumen de datos (datagramas) y al número de fuentes conectadas (buses, estaciones, sensores, etc.). Debe poder procesar los nuevos eventos sin afectar la disponibilidad del sistema y asegurar que los resultados analíticos reflejen la realidad operativa.
12	El sistema debe ser escalable en el procesamiento de los eventos que son generados por los buses que se ponen a funcionar diariamente.
13	El sistema debe ofrecer servicios que permitan a los ciudadanos, empresas o entidades públicas consultar información útil sobre el estado del sistema de transporte y las estimaciones de tiempo promedio de viaje. Esta información debe poder accederse mediante interfaces o servicios interoperables, con una presentación adecuada y confiable.
14	En la fase de análisis debe plantearse el diagrama de transición de estados tanto de los eventos como de los controladores. Estos diagramas de transición de estados deben revisarse y validarse con el cliente antes de pasar al diseño.

Su misión es diseñar e implementar la arquitectura del subsistema que involucre todos los elementos necesarios para automatizar el cálculo de la velocidad promedio de todos los arcos del grafo que representa las rutas del sistema. Este cálculo se debe hacer en dos escenarios:

1. Usando los datos históricos de los eventos de posición GPS que han enviado los buses.
2. (Opcional-Bono) Usando los datos de "streaming" para actualizar los valores en tiempo "real"

Para lo anterior, y teniendo en cuenta la clasificación de los requerimientos, deben desarrollar y entregar:

- A. (10%) Árbol de particionamiento refinado hasta el segundo nivel de profundidad, para el escenario señalado.
- B. (20%) Especificaciones en formato bicolumnar que cubran el escenario señalado, incluyendo sus sub-casos.
- C. (35%) Diagrama de deployment (diseño arquitectónico completo y consistente con las respuestas anteriores) que soporte el escenario señalado. Debe explicar las estructuras distribuidas de procesamiento y de almacenamiento, y resaltar sobre el diagrama de deployment los patrones de diseño introducidos para satisfacer el driver de arquitectura de performance (escalabilidad, latencia y throughput).
- D. (20%) Implementación y deployment de la solución: Debe mostrar el despliegue de la solución y probarla para 3 archivos de datagramas de tamaños normalizados en una escala logarítmica (e.g., 1.000.000, 10.000.000, 100.000.000). Los tamaños los escogen los integrantes del equipo. Deben medir el tiempo que toma procesarlos y el número de nodos de procesamiento.
- E. (15%) Informe de experimentos y ejecución con por lo menos 4 nodos de procesamiento, incluyendo gráfico para determinar el punto de corte a partir del cual es necesario distribuir el procesamiento.
- D. (15%) Procesar los datos en tiempo real (bonus).