

Marco Antonio Pérez
Jose Manuel Ossorio
Fabio Andrés Mejía
Nelson David Quiñones
Juan David Aguirre

Método de la ingeniería

Fase 1: identificando el problema:

- Descripción del contexto del problema:

En la ciudad de Cali existe un sistema de transporte llamado Masivo Integrado de Occidente, o como comúnmente se le conoce, MIO. Dicho sistema da acceso al 75% de toda el área metropolitana de la ciudad de Cali por medio de numerosas rutas cuya disponibilidad varía según la temporada. La empresa gestora de la planeación de dichas rutas es Metro Cali, la cual crea diferentes planes para las rutas que seguirán los buses cada día. Más específicamente, Metro Cali se basa en temporadas escolares, festividades, y grandes eventos temporales que ocurren en la ciudad de Cali para planear qué rutas estarán disponibles y de cuantos buses dispondrá cada ruta.

- Descripción del problema:
 - a. MetroCali desea poder visualizar cómo se comportan los buses para rutas definidas de tal forma que puedan obtener información.
- Requerimientos funcionales:
 1. El programa permitirá la visualización de un mapa de la ciudad de Cali.
 2. Mostrar en el mapa todas las paradas de los buses del MIO.
 3. Visualizar el movimiento de los buses del MIO activos a lo largo de sus rutas (simulación).
 4. Modificar el tiempo de visualización de la simulación
 5. Filtrar los datos que se muestran durante la simulación (rutas, zona, tipo de bus, etc).
 6. Analizar y comparar la el tiempo que ha recorrido un bus sobre un arco determinado con la tiempo promedio que debería recorrer ese mismo bus en ese mismo arco.
- Requerimientos no funcionales:
 1. Toda la información debe ser almacenada en una base de datos.
 2. La base de datos usada debe correr en un servidor externo y no localmente. Es claro que durante el desarrollo es necesario que el diseño lógico se debe realizar localmente para posteriormente subir el esquema conforme la lectura, modificación y adición de información sea estable.
 3. El manejo de información debe hacerse bajo protocolos de seguridad como SSL.

4. La implementación se debe realizar en C#

Fase 2: recopilación de la información necesaria:

El sistema de transporte MIO está compuesto de líneas de ruta las cuales reciben un nombre, como por ejemplo E21, estas especifican además del nombre de la ruta, su parada inicial y final del recorrido, como por ejemplo la ruta E21 que va desde menga hasta universidades. Además, la línea de ruta incluye una descripción de la línea.

Las líneas tienen varias tareas, las cuales se encargan de especificar el plan y el horario que se va a seguir. Para esto, cada tarea contiene la referencia de la línea que la está siguiendo, el plan y el programa asignado. Las tareas a su vez pueden tener varios viajes, los cuales muestran cada uno de los tramos que el bus siguió para completar la tarea asignada. Estos viajes contienen las áreas por las cuales pasa el bus, que a su vez contienen diversas paradas. De las paradas se especifican su ubicación en latitud y longitud y su nombre.

Estado del arte:

Mio_App:

El equipo de MetroCali lanzó su propia aplicación el 5 de noviembre de 2018 desde la cual los usuarios pueden:

- Planear un viaje
- Visualizar las rutas en tiempo real
- Consultar los puntos de recarga
- Consultar su saldo

Rappi:

Rappi es un *marketplace* en línea lanzado el 6 de junio de 2015 que permite a sus usuarios:

- Comprar desde casa productos (sea comida, medicina, ropa, etc) que serán comprados y entregados por un tercero, conocido internamente como un *rappitendero*
- Visualizar la posición en tiempo real del *rappitendero*

EasyTaxi:

EasyTaxi o Cabify es una aplicación en la que cualquier usuario puede solicitar un servicio de transporte, sea este un taxi o un carro privado. Entre sus funcionalidades encontramos:

- Visualizar en tiempo real de la posición del automóvil que prestará el servicio
- Visualizar un tiempo esperado que le tomará al conductor en llegar al lugar en el que se solicitó el servicio

Fase 3: soluciones creativas:

Solución del problema:

Utilizamos el método relación forzada para generar estas ideas. Los conceptos a cruzar son:

- Líneas
- Áreas
- Arcos

- Colores
- Círculos
- Rectángulos
- Listas
- Buses
- Comportamiento esperado
- Comportamiento observado
- Marcadores o marker
- Tipo de bus
- Indicador de velocidad

Obtenemos las siguientes relaciones con detalles:

- *Dibujar áreas circulares* de distintos colores denotando su relación con el *comportamiento esperado*.
- *Dibujar áreas rectangulares* de distintos colores denotando su relación con el *comportamiento esperado*.
- *Dibujar arcos* de distintos colores denotando su relación con el *comportamiento esperado*.
- Modificar los *colores* de los *marcadores* que representan buses dependiendo de su comportamiento comparado con el *esperado*.
- Modificar los *colores* de los *marcadores* que representan buses dependiendo de su comportamiento observado y su indicador de velocidad.
- Modificar los *colores* de los *marcadores* que representan *estaciones* dependiendo de su comportamiento observado y su indicador de velocidad.
- *Listar* los *arcos* donde se cumple (con un error del 5% por ejemplo) con el *comportamiento esperado*.
- *Listar* las *estaciones* donde se cumple (con un error del 5% por ejemplo) con un mismo indicador de velocidad.
- *Listar* los *Buses* segmentados por el tipo, que cumplan (con un error del 5% por ejemplo) con el *comportamiento esperado*.
- *Listar* los *Buses* segmentados por el tipo que cumplan (con un error del 5% por ejemplo) con un mismo indicador de velocidad.
- Mostrar el *comportamiento esperado* con distintos *colores* relacionando su comportamiento con lo esperado.

Almacenamiento y lectura de datos.

Un punto externo al problema, pero con amplio tiempo necesario de análisis, será la selección de la base de datos. Antes de avanzar con el análisis propio del problema y sus posibles soluciones, la manera de almacenar la información será definida previamente. Las siguientes bases de datos se presentan como opciones viables:

- Bases de datos relacionales:
 - Guardar y consultar información haciendo uso de MySQL
 - Guardar y consultar información haciendo uso de Oracle
 - Guardar y consultar información haciendo uso de Microsoft SQL Server
- Bases de datos no relacionales:
 - Guardar y consultar información haciendo uso de MongoDB
 - Guardar y consultar información haciendo uso de Firebase

Fase 4: transición a diseños preliminares:

Es crucial para el proyecto la elección de una base de datos que funcione de manera adecuada para nuestras necesidades y que sea relativamente fácil de implementar. De la elección de la mejor solución para la base de datos se hablará en la siguiente fase, en donde se analizarán todas según ciertos criterios definidos por los integrantes del equipo.

Por otro lado, en cuanto a la visualización de los datos y el análisis de los mismos, nos puede interesar representar si los buses están realizando sus recorridos dentro del tiempo esperado, colorear un arco o una zona de verde o rojo si los buses están cumpliendo con lo planeado o no, respectivamente, resulta siendo una solución a este aspecto. Ahora, también se requiere precisión en visualizar qué buses o líneas están cumpliendo con la planeación, por lo que “Arcos de distintos *colores* denotando su relación con el *comportamiento esperado*” sería un mejor acercamiento que “Áreas circulares o rectangulares de distintos colores denotando su relación con el *comportamiento esperado*”. Sin embargo, esta última idea puede ser útil al momento de ver cómo se comporta una zona en particular, por lo que no debe ser descartada inmediatamente.

En términos de desempeño del programa, sospechamos que solo el hecho de visualizar todas las paradas a la vez en el mapa causará que la interfaz funcione de manera aletargada y no muy apacible para el usuario al momento de usar funcionalidades como zoom o desplazarse dentro del mapa con el mouse.

Debido a la anterior conjunción, hemos decidido tomar otra ruta para visualizar los datos con las paradas. Nuestra idea es que el programa solo grafique las paradas (en este caso se habla de paradas pero puede aplicar para el resto de objetos a visualizar) que están en un radio de visión definido. Por ejemplo, una opción sería bloquear el zoom del mapa a cierto valor y solo graficar lo que se alcanza a percibir en el área que el usuario está viendo en un momento específico. Así, el mapa no tendría tantos *markers* (en este caso *markers* se usa para representar las paradas, pero podría ser cualquier otro de los elementos disponibles en GMap usado para representar otra entidad visual de los datos) para mostrar y se ejecutaría de manera mucho más fluida, ergo, sería una experiencia mucho más placentera y amigable para el usuario.

Otra consideración que se tuvo en cuenta pensando en la eficiencia de la aplicación fue el enfoque que le podemos dar a la visualización del desempeño de los buses y estaciones. Así fue como entró el indicador de velocidad como una posible opción para optimizar el programa y dar información útil sobre el desempeño del sistema. El indicador de velocidad representa la velocidad en la que la que viaja un bus o la velocidad promedio con la que llegan los buses a una estación en particular. Lo anterior permite entonces entender las dinámicas entre los buses y las estaciones, resolviendo así el problema.

Diagrama de clases

Fase 5: Evaluación y selección de la mejor solución:

Selección de base de datos.

Con la finalidad de seleccionar la base de datos que mejor se adecúe a las necesidades, capacidades y limitaciones del grupo tendremos en cuenta los siguientes campos:

- Curva de aprendizaje, haciendo referencia a la necesidad de tiempo considerable para el aprendizaje y mejor manejo de la tecnología:
 - 1 - curva de aprendizaje muy alta. Esto conlleva aprender nuevas tecnologías y formas de guardar la información.
 - 2 - curva de aprendizaje media. Esto conlleva solamente un aprendizaje de nuevas tecnologías.
 - 3 - curva de aprendizaje baja. Tanto la tecnología como la manera en que se guardan son previamente conocidas por los integrantes del grupo.
- Manejo de altos volúmenes de datos:
 - 1 - *Queries* ineficientes cuando la base de datos maneja gran volumen de datos.
 - 2 - *Queries* eficientes cuando la base de datos maneja gran volumen de datos.
- Integración con Visual Studio, es decir, qué tan sencillo es conectar la base de datos con la aplicación:
 - 1 - Conexiones completamente por código propio de los desarrolladores.
 - 2 - Conexiones a través de plugins presentes en Visual Studio.
 - 3 - Soporte nativo de la base de datos.
- Costos
 - 1 - Pago por soporte y uso de funcionalidades básicas.
 - 2 - Pago por soporte y funciones extra.
 - 3 - Pago por soporte.
 - 4 - Gratis sea por convenio o por naturaleza de la tecnología.

Base de datos	Criterio 1	Criterio 2	Criterio 3	Criterio 4	Total
MongoDB	1	2	2	2	7
SQL Server	2	2	3	4	11
MySQL	3	1	2	2	8
Oracle	3	2	3	4	12
Firebase	1	2	1	2	6

Debido a los resultados, Oracle es la opción más viable y con la cual se llevará a cabo el proyecto. Oracle es ideal pues por convenio su versión *Enterprise* es totalmente gratis. Por otra parte, la curva de aprendizaje a pesar de ser una tecnología nueva para los integrantes del grupo, no es tan alta como para SQL Server puesto que Oracle es enseñada en los cursos relacionados a modelamiento y bases de datos.

Selección de solución:

Con la finalidad de seleccionar la estrategia de visualización y análisis que mejor se adecúe a las necesidades, capacidades y limitaciones del grupo tendremos en cuenta los siguientes campos:

- Eficiencia temporal del análisis:
 - 1 - Se maneja una cantidad de operaciones acotadas por una función $O(n^2)$.
 - 2 - Se maneja una cantidad de operaciones acotadas por una función $O(n \cdot \lg(n))$.
 - 3 - Se maneja una cantidad de operaciones acotadas por una función $O(n+m)$.
- Manejo de volúmenes de datos para realizar el análisis:
 - 1 - Se necesitan los arcos, las estaciones y los buses.
 - 2 - Se necesitan las estaciones y los buses.
- Eficiencia temporal para visualizar los resultados:
 - 1 - Se maneja una cantidad de operaciones que dependen de los arcos y los buses.
 - 2 - Se maneja una cantidad de operaciones que dependen de las estaciones y los buses.

Para la elección de los criterios se debe tener en cuenta que la cantidad de arcos es mayor a la cantidad de estaciones, y la cantidad de estaciones es mayor que la cantidad de buses.

Conceptos de visualizaci	Criterio 1	Criterio 2	Criterio 3	Total
Arcos	1	1	1	3
Indicador de Velocidad	3	2	2	7

Fase 6: Preparación de reportes y especificaciones:

Los archivos PNG de los diagramas pueden encontrarse en la carpeta Docs del repositorio

Modelo Lógico de la base de datos:

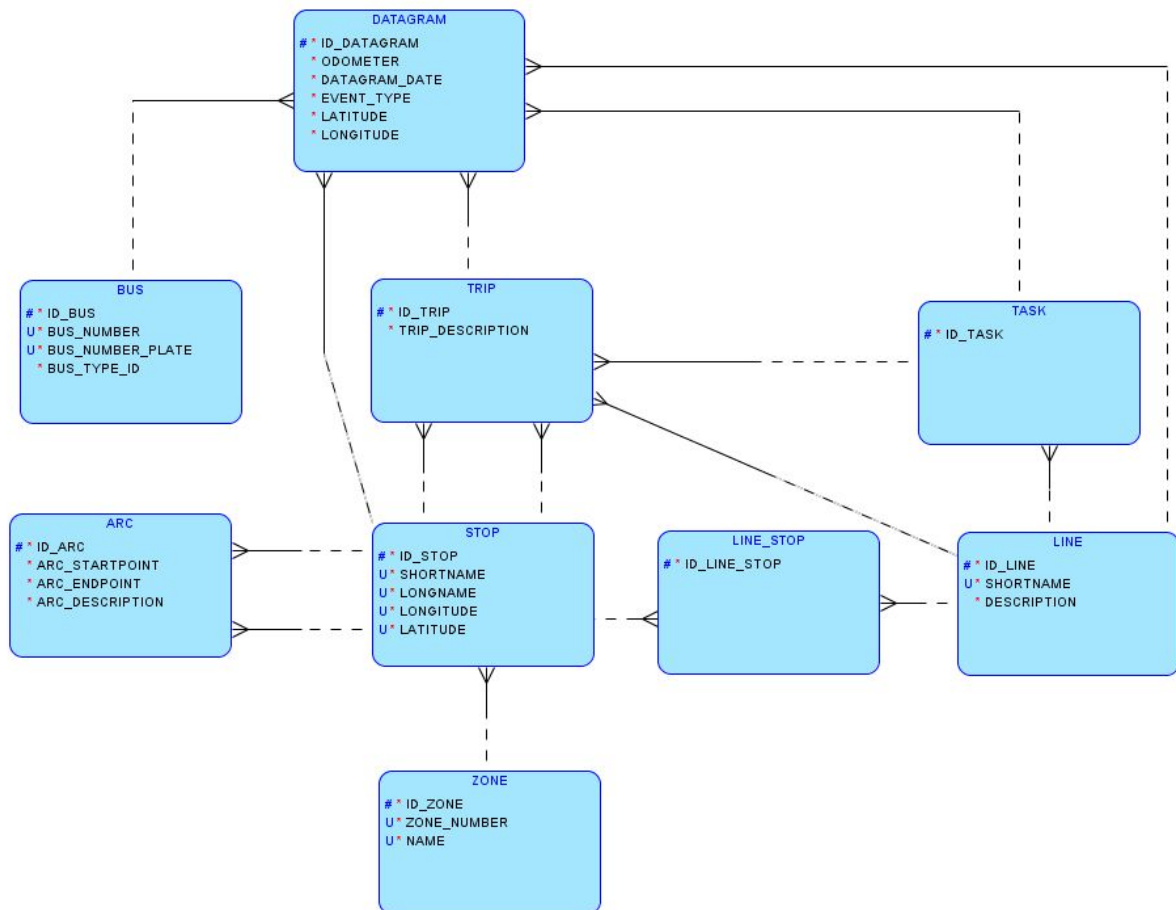


Diagrama de clases:

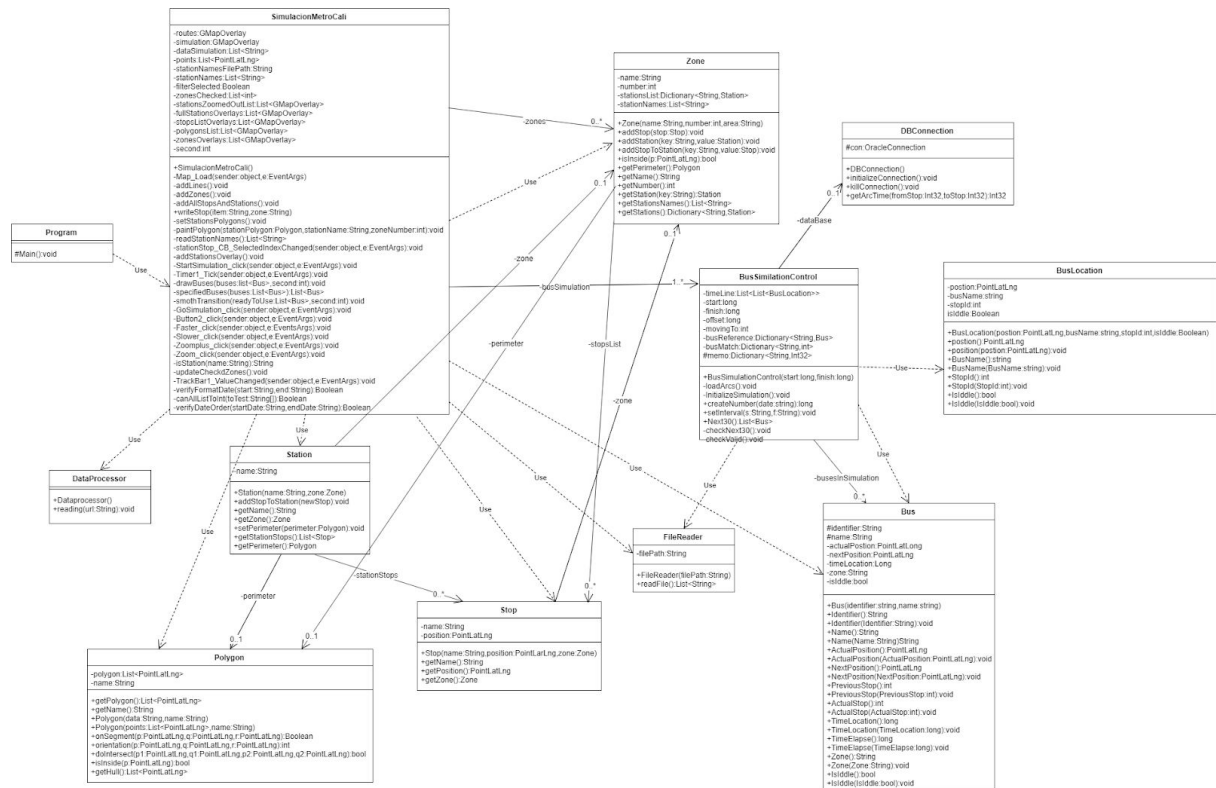


Diagrama de secuencia del método setStationsPolygons

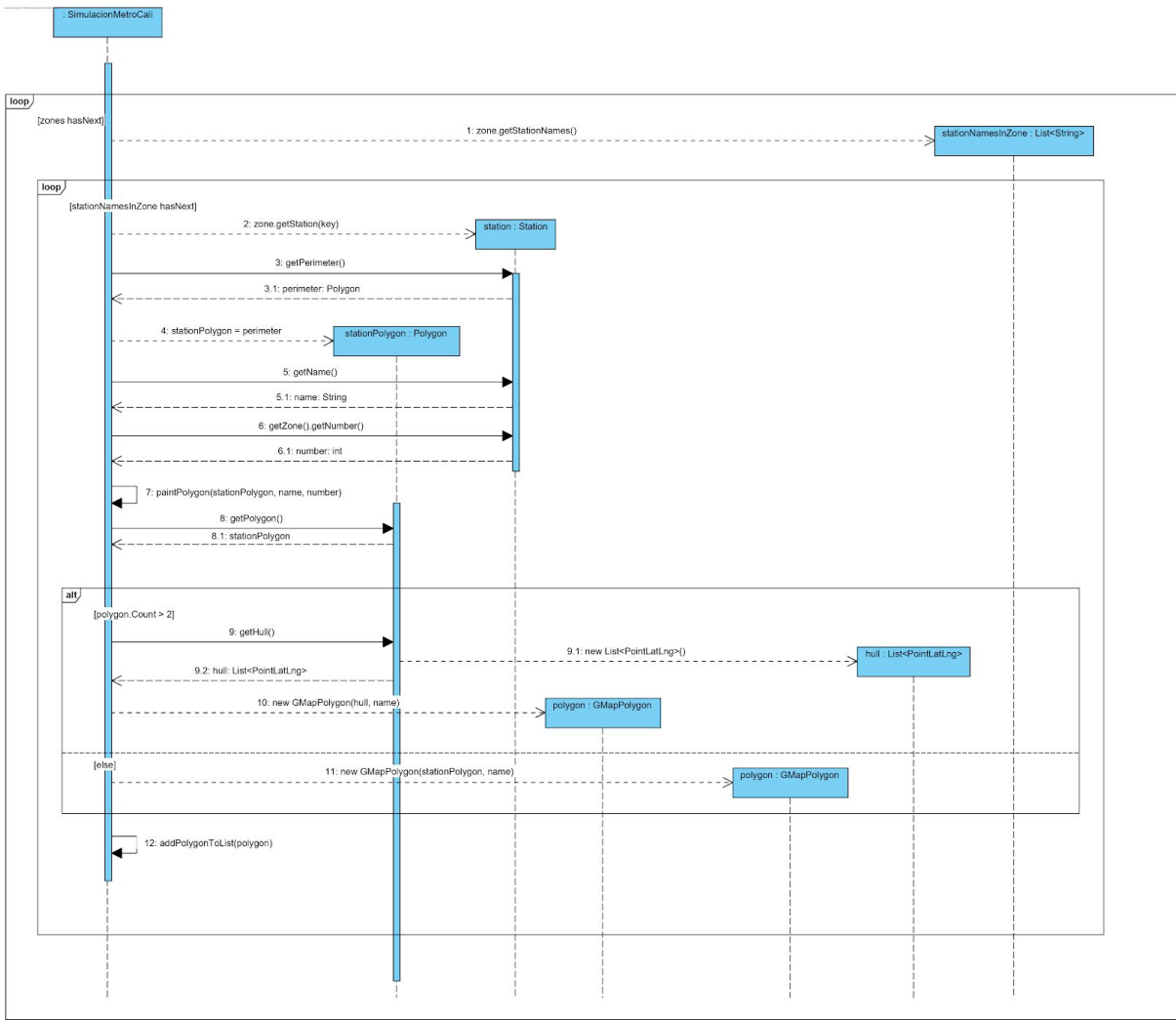


Diagrama de secuencia de la transición de un sólo bus

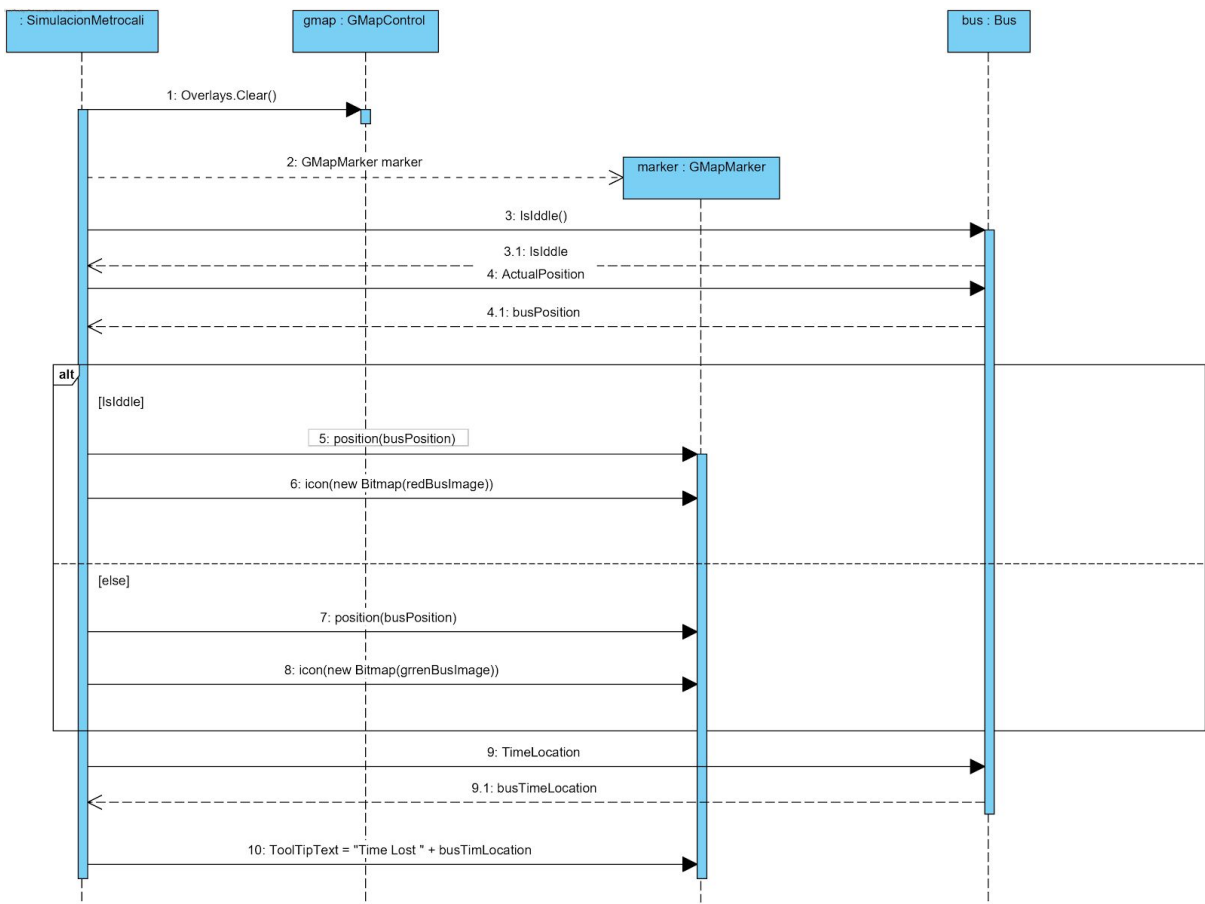


Diagrama de secuencia startSimulation_Click:

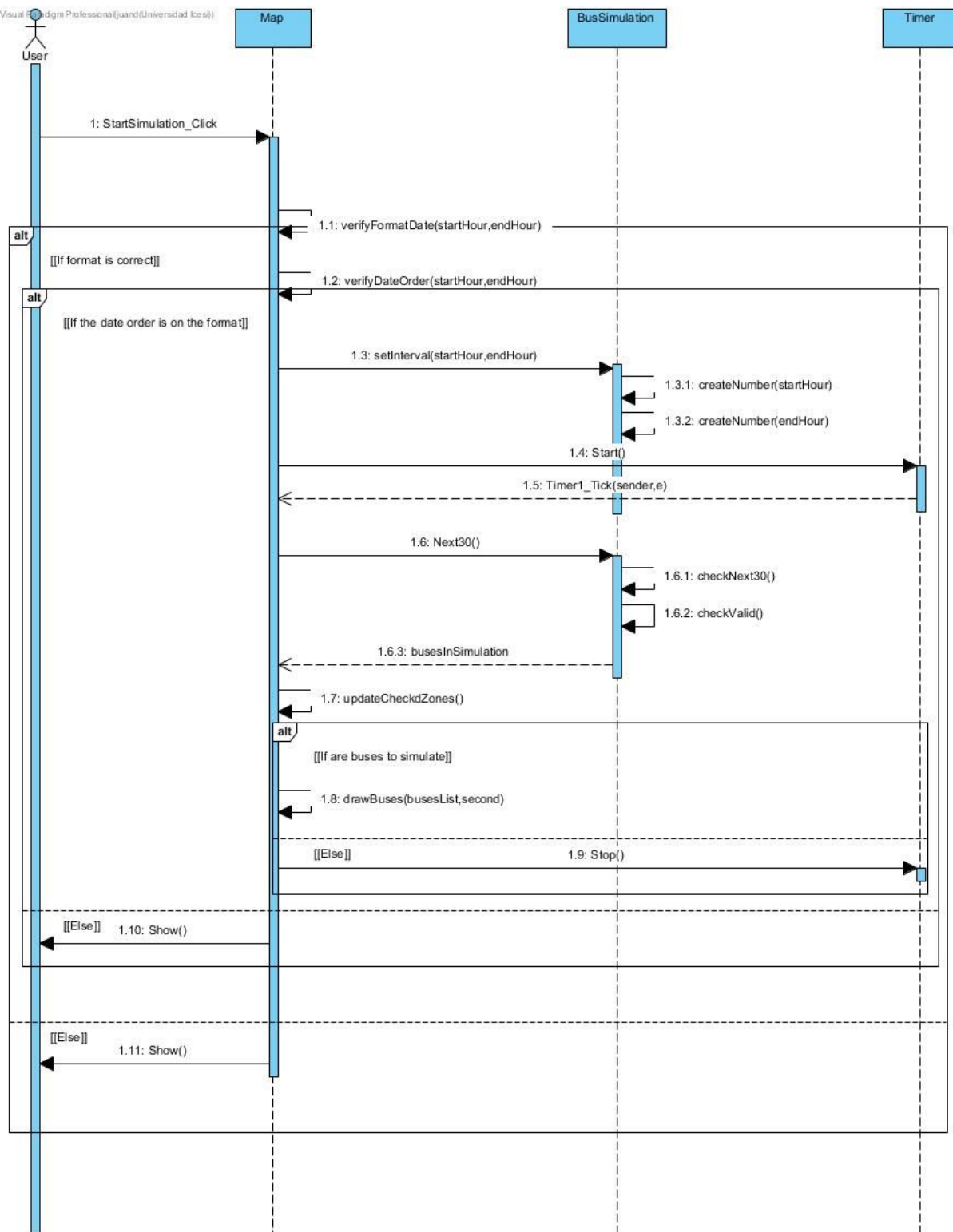


Diagrama de objetos:

