Universidad de Concepción
Facultad de Ingeniería
Dpto. Ingeniería Civil Informática y Ciencias de la Computación

Modelo de Integración de Datos de Objetos en Movimiento

Tesis para optar al grado de Magíster en Ciencias de la Computación

Meraioth Ulloa Salazar

Profesor guía: Andrea Rodríguez Tastets

28 de abril de 2020, Concepción, Chile

Índice

1. Introducción

En la actualidad y gracias al avance de la tecnología, los servicios basados en localización (location-based services) han ganado gran popularidad. Estos servicios utilizan diversos dispositivos tales como GPS y RFID, lo que causa que las fuentes y los formatos no sean uniformes. Se tiene, por ejemplo, los datos recolectados de distintas fuentes del sistema de transporte público. Tarjetas inteligentes (smart cards) que registran la entrada de usuarios al transporte público y en algunos casos también su salida [19]. En este mismo dominio, los buses del transporte público con dispositivos GPS registran cada cierta unidad de tiempo la posición de un servicio en particular. En forma abstracta, el sistema de transporte se compone de objetos (usuario o buses) cuyas localizaciones varían en el tiempo y sus movimientos se describen como trayectorias. Estas son representadas como secuencias de posiciones ordenadas temporalmente.

En respuesta al uso de estos datos han surgido sistemas que representan, almacenan y procesan datos de objetos en movimiento, sistemas conocidos como bases de datos de objetos en movimiento (MOD, por su acrónimo en inglés) [6, 7, 9, 26]. En los últimos años se ha desarrollado bastante trabajo en modelos y estructuras (índices) de datos de objetos en movimiento [3, 4, 8, 21–23, 27] que mejoran el rendimiento de consultas. A su vez, se han realizado investigaciones sobre el almacenamiento masivo de datos en lo llamado almacén de datos de trayectorias o TDW (trajectory data wharehouse) [16, 18, 20], los que soportan procesamiento analítico en línea u OLAP (Online Analytical Processing) [20] y herramientas de minería de datos. Los TDW son capaces de responder consultas de agregaciones considerando múltiples dimensiones y precalcular algunas consultas (para optimizar consultas) a diferencia de una base de datos regular que siempre está siendo actualizada por lo cual todas las agregaciones son en tiempo de consulta.

A pesar de los trabajos realizados en el contexto de bases de datos de objetos en movimiento, un problema que aún existente es la integración de datos provenientes de diferentes fuentes y con diferentes formas de representación. Tomando como ejemplo un sistema de transporte público donde existen 3 tipos de fuentes de datos: (i) GTFS (General Transit Feed Specification), (ii) datos GPS de los buses y (iii) datos de las tarjetas inteligentes que contiene dónde y cuándo un usuario subió y bajó de un bus. Los GTFS y los datos GPS de buses describen la oferta, mientras que el uso de tarjetas inteligentes caracterizan la demanda del sistema de transporte público.

Se considera que un modelo integrado de datos de objetos en movimiento es útil para:

- Responder consultas que combinan diferentes fuentes, consultas tales como ¿Cuántas personas usan el sistema en un área geográfica?
- Almacenar eficientemente los datos, evitando duplicación de estos. Por ejemplo, no se necesita almacenar todos los puntos donde la gente se encuentra ya que el bus donde viajan mantiene esa información.

Una visión integrada del sistema debería ser capaz de relacionar los datos acerca de los viajes de los usuarios en los respectivos servicios. Incluso aún más, debería ser capaz de reflejar que los usuarios cambian de servicio y de tipo de transporte en paraderos, de tal forma que su viaje es una secuencia de etapas en un medio de transporte en particular (bus o tren), registrando el momento de subida y/o bajada del sistema. Por otro lado, una trayectoria de un servicio de buses es una secuencia temporal-ordenada de paraderos, la cual puede ser mapeada a una red. También un servicio es definido en la fuente GTFS como una secuencia de paraderos por el cual pasa, donde es posible distinguir las paradas donde realmente estuvo un tiempo y otras donde solo pasó por ellas. En una visión integrada de los datos, para un usuario que aborda un bus determinado, los paraderos de subida y bajada deben formar parte del viaje realizado por dicho servicio.

Dado lo anterior, preguntas interesantes a considerar son:

1. ¿Cuántos recorridos realiza un servicio de buses durante un día?
2. ¿Cuántos buses circulan por Santiago entre ciertas horas?
3. ¿Cuántos buses pasan por una comuna particular durante un día?
4. ¿Cuántas líneas intersecan un recorrido dado?
5. ¿Cuáles buses pasan por un paradero en particular en un instante?
6. ¿Qué viaje realiza una persona? o ¿Cuántos viajes realiza en un día?

Aunque existen modelos de datos para objetos en movimiento, hasta donde se conoce en el estado del arte no existe un modelo integrado que permita manejar diferentes formas de representar objetos en movimiento. Este trabajo trata de cerrar esa brecha, por lo cual apunta a proponer e implementar un modelo de datos que integre distintas representaciones en el contexto de datos de transporte público.

1.1. Hipótesis

Un nuevo modelo de integración de datos de objetos en movimiento permite un acceso en forma homogénea de distintos tipos de representaciones, mantiene el mismo poder de expresividad que los modelos actuales y su implementación es eficiente en el uso de espacio.

1.2. Objetivo general

Crear un modelo integrado de datos para objetos en movimiento históricos, con sus respectivos tipos de datos y operadores para el correcto almacenamiento y manipulación de estos.

1.3. Objetivos específicos

- Estudiar un modelo abstracto de datos, que permita el uso de datos de diferente tipo y fuente de origen.
- Diseñar e implementar tipos de datos que puedan representar objetos en movimiento en distintos dominios espaciales.
- Diseñar e implementar algoritmos para las funciones de mapeo, con el fin de representar una trayectoria en distintos dominios espaciales.
- Evaluar de forma experimental los algoritmos implementados utilizando datos reales y considerando su costo computacional en tiempo y espacio.
- Evaluar el modelo en términos de costo computacional y capacidad de expresividad en consultas rango temporal y espacial.

2. Conceptos Previos y Trabajo Relacionado

Existe una gran cantidad de trabajo en modelos de datos para objetos en movimiento [6, 7, 9, 10, 13, 25, 26, 30]. Básicamente, un modelo de datos de objetos en movimiento representa los cambios continuos de la ubicación de un objeto a través del tiempo, lo que es llamado trayectoria del objeto. En un nivel abstracto, las trayectorias han sido pensadas como tipos de datos genéricos definidos por una función de mapeo desde el tiempo a espacio [10].

Los datos de objetos en movimiento se distinguen según: la representación de la ubicación, la información contextual o ambiental donde el movimiento toma lugar, la dimensión del tiempo que puede ser continua o discreta, como también el nivel de abstracción o granularidad sobre el cual las trayectorias son descritas [31].

Como consecuencia de la diversidad en los tipos de trayectorias, existen diversos modelos que representan tipos de trayectorias particulares haciendo uso de definiciones algebraicas de tipos de datos y de operadores para estos.

A continuación se presentan las directrices de los modelos de datos para objetos en movimiento, los que se clasifican en dos categorías: (1) modelos para el movimiento histórico de objetos en movimiento, los cuales describen la evolución de un objeto en el tiempo en base a su historial de movimientos y (2) modelos para el movimiento futuro o predictivo de objetos en movimiento, los cuales representan la evolución de un objeto en el tiempo considerando la predicción de sus movimientos futuros.

Dentro del primer grupo nace una segunda clasificación, en primer lugar está el modelo que representa trayectorias euclidianas (ver Sección 2.1.1), estas son una secuencia de puntos GPS representados por un tipo de dato apropiado. Para objetos en movimiento, por lo general se utilizan geometrías de tipo punto, las cuales tienen asociadas un par de coordenadas cartesianas x e y para indicar su posición, estos no poseen restricciones sobre sus movimientos en el espacio, por ejemplo, el movimiento de animales o personas en áreas libre. En la segunda clasificación está el modelo que representa las trayectorias embebidas en redes (ver Sección 2.1.2), estas son una sucesión de ubicaciones en la red, ordenadas con respecto al tiempo, se limita a los objetos en cuestión a moverse sobre redes definidas. Aquí se encuentra el caso de medios de transporte como automóviles, buses, trenes, entre otros, cuyos movimientos deben realizarse dentro de caminos, carreteras o rutas determinadas.

2.1. Modelos de datos para el movimiento histórico de objetos

Dentro de esta clasificación existen dos tipos aún más específicos de modelos de datos para el movimiento histórico de objetos; para objetos en movimiento en espacio libre y para objetos en movimiento restringidos a redes.

2.1.1. Modelos de datos para el movimiento histórico de objetos en espacio libre

En esta subcategoría se aborda la representación de datos de objetos cuyo movimiento se realiza de forma libre en el espacio [6, 10].

Para comenzar, en [6] se definen tipos de datos para representar el movimiento de los objetos. El movimiento histórico de los objetos se modela como una función que toma un elemento espacial y le añade una dimensión de tiempo, obteniendo distintos tipos de datos móviles. Para esto se utiliza un constructor de tipo, τ, el cual transforma un tipo de dato α a un tipo de dato $\tau(\alpha)$, de la siguiente forma:

$$\tau(\alpha) = time \rightarrow \alpha$$

Al aplicar la función sobre un tipo de dato espacial, como punto (*point*), línea (*line*) o región (*region*), se genera el tipo de dato móvil correspondiente, en este caso, punto móvil (*mpoint*), línea móvil (*mline*) y región móvil (*mregion*) respectivamente, esto se señala a continuación:

$$mpoint : \tau(point) = time \rightarrow point$$
$$mline : \tau(line) = time \rightarrow line$$
$$mregion : \tau(region) = time \rightarrow region$$

Luego, en [10] se define un conjunto de operadores sobre los tipos de datos temporales antes mencionados. Algunos operadores son:

$$locations : mpoint \rightarrow points$$
$$trajectory : mpoint \rightarrow line$$
$$routes : mline \rightarrow line$$

Donde *locations* retorna un valor *points*, mientras que *trajectory* y *routes* retornan un valor *line*, indicando las distintas formas en que se puede representar la proyección de un *mpoint*.

2.1.2. Modelos de datos para el movimiento histórico de objetos restringidos a redes

Las trayectorias de los objetos en movimiento pueden estar restringidas a redes espaciales específicas. De aquí se concluye que estas redes deben formar parte del modelo de datos utilizado. En [12] se plantea la representación de una red como un conjunto de rutas y de uniones entre estas rutas. Se define entonces una ruta de la siguiente forma:

$$Route = \{(id, l, c, kind, start) | id \in int, l \in real, c \in line, \\ kind \in \{simple, dual\}, start \in \{smaller, larger\}\}$$

Donde *id* corresponde a un identificador de tipo entero, l es la longitud de la curva, siendo un dato de tipo real, c describe la geometría en el plano que corresponde a un tipo de dato *line*, *kind* indica el tipo de ruta, el cual puede ser simple o doble y *start* sirve para señalar si la ruta comienza en el punto lexicográficamente más pequeño o más grande.

Luego, para indicar una posición en una ruta, se define una medida de ruta (*route measure*), la cual consiste en el identificador de una ruta y un valor real que denota una posición en la curva. Se define una medida de ruta en R como:

$$RMeas(R) = \{(rid, d) | rid \in int, d \in real, \exists (rid, l, c, k, s) \in R \text{ tal que } 0 \leq d \leq l\}$$

Además, debido a que una ruta puede ser simple o doble, se añade el valor *side*, para indicar el lado de la ruta. Así, se define entonces la ubicación de ruta en R (*route location*) como:

$$RLoc(R) = \{(rid, d, side) | (rid, d) \in RMeas(R), side \in Side, \\ \text{para } (rid, l, c, kind, start) \in R : kind = simple \leftrightarrow side = none\}$$

Para representar la intersección de dos rutas se define una unión en R (*junction*), como una tripleta que consta de dos medidas de ruta en R (una de cada ruta intersecada) y un código de conectividad, el cual consiste en un valor entero que indica el movimiento en la unión.

$$Junction(R) = \{(rm1, rm2, cc) | rm1, rm2, \in RMeas(R), \\ rm1 = (r1, d1), rm2 = (r2, d2), r1 \neq r2, cc \in int\}$$

Se define una red como un conjunto de rutas y uniones entre ellas y se representa de la siguiente forma:

$$N = (R, J)$$

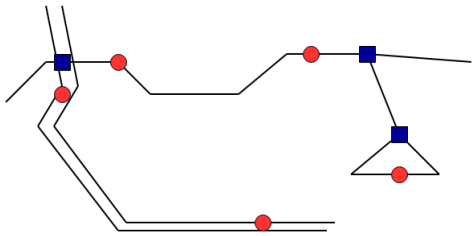

Figura 1: Ejemplo de una red, en cuadrado junctions, en líneas simples routes simples, en líneas dobles routes dual, en círculos route locations. Fuente: elaboración propia.

Todo lo anterior corresponde a la parte espacial del modelo propuesto en [12], para la parte temporal se utiliza la representación basada en porciones o intervalos de tiempo para objetos en movimiento.

Supongamos que un objeto en movimiento en una red, empieza en t_0 y termina en t_n, para representar este objeto según el modelo, se debe entonces dividir el tiempo en intervalos disjuntos entre si, para cumplir con la restricción que ningún objeto puede estar en distintos lugares en un mismo tiempo como se observa en la figura 2. Para responder a qué tiempo corresponde x posición, dado que solo se tiene intervalos de tiempo, el modelo propone ocupar funciones simples, por ejemplo para puntos en movimientos se ocupan funciones lineales para calcular posiciones (considerando velocidad constante entre intervalos de tiempo).

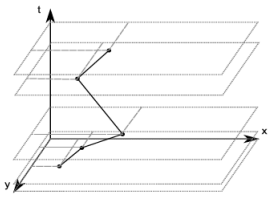

Figura 2: Ejemplo de un objeto en movimiento con representación time-sliced en espacio libre. Fuente : [5]

$$GPoint = \{(rid, pos, side) | rid \in int; pos \in real; side \in (up, down, none)\}$$

Finalmente una trayectoria es representada como una secuencia ordenada temporalmente de *ugpoint*, tipo de dato que es compuesto por dos *gpoint* y dos marcas de tiempo.

$$UGPoint = \{(rid, t1, t2, pos1, pos2, side) | rid \in int; t1, t2 \in Instant; pos1, pos2 \in real; side \in (up, down, none)\}$$

2.2. Modelos de datos para el movimiento predictivo de objetos

Entre los modelos de datos que se enfocan en el movimiento predictivo de objetos se encuentra MOST (Moving Objects Spatio-Temporal) [25].

La idea fundamental es introducir los llamados atributos dinámicos que cambian sus valores automáticamente con el tiempo. No todos los tipos de atributos son elegibles para ser dinámicos, el tipo de atributo debe tener un valor 0 y una operación de suma. La dinámica viene dada por funciones lineales que describen vectores de movimiento y evitan frecuentes actualizaciones de bases de datos. Ejemplos de esto son los tipos: entero y real dinámico. Lamentablemente, no existe un concepto de tipos de datos espaciales dinámicos, de modo que la única opción para representar un moving point es modelarlo como un par (x: real dinámico, y: real dinámico). Las líneas o regiones dinámicas no se pueden modelar, y no hay ningún concepto de tipos de datos espacio temporales disponibles. Si una consulta se refiere a un atributo dinámico A, su valor dinámico se entiende y se utiliza en la evaluación. Por lo tanto, el resultado depende del momento en que se emite la consulta. Si tal consulta se reevalúa en cada tic del reloj, esta consulta se llama continua. Para este modelo se desarrolló también un lenguaje de consulta tipo SQL, llamado Future Temporal Logic (FTL), incluyendo operadores como until y nexttime.

Como un modelo más reciente se encuentra FuMMO (Future Movements of Moving Objects) [24], el cual no genera predicciones sobre los movimientos futuros de objetos, sino que los modela. Se utilizan los conceptos de rango espacial y rango espacio-temporal, que corresponden a la región predicha dentro de la cual se puede encontrar el movimiento futuro del objeto y al volumen predicho dentro del cual se puede encontrar el movimiento futuro del objeto en un tiempo determinado, respectivamente. Este rango, espacial o espacio-temporal, en conjunto con una distribución de probabilidad específica generan el modelo de predicción.

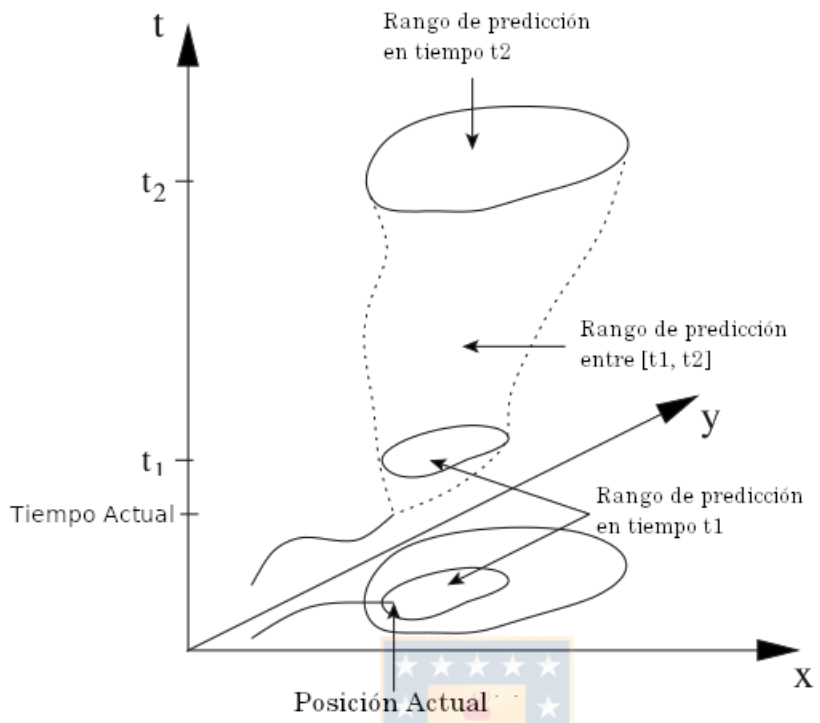

Figura 3: Predicción según modelo FuMMO. Fuente [24]

Estos modelos de movimiento predictivo están fuera del alcance de este trabajo ya que solo se enfocará en modelos de movimiento histórico.

2.3. Integración de datos

Uno de los principales objetivos de los sistemas de bases de datos heterogéneos es lograr la interoperabilidad de los datos. En este sentido existe una gran cantidad de investigaciones [1, 2, 17, 28] y distintas alternativas de solución, desde sistemas de bases de datos federadas, sistemas multibase de datos, hasta bases de datos distribuidas.

En las últimas dos décadas, las investigaciones en esta área comienzan a tomar relevancia, surgiendo varios trabajos sobre la integración de datos provenientes de distintas fuentes y la generación de una vista unificada para los usuarios [14,15,29].

Estas investigaciones se enfocan en el manejo de bases de datos clásicas, sin embargo, la cantidad de datos de tipo espacial y espacio-temporal que se generan en la actualidad han hecho que en los últimos años los esfuerzos se centren en el

estudio de sistemas específicos para su integración y de los distintos factores que deben ser considerados para la obtención de información coherente.

Un reciente modelo de datos genérico para objetos en movimiento [31] representa el movimiento en diferentes ambientes y distintos medios de transporte. Este modelo considera cinco tipos de ambientes: redes de transporte público, espacio libre, carreteras de redes, exterior basado en regiones e interior.

La representación de la ubicación de un objeto en movimiento tiene dos componentes :

- Ubicaciones genéricas, modeladas como un conjunto de tuplas de la forma (oid,(loc1,loc2)), donde oid es un identificador del tipo de ambiente y (loc1,loc2) representa una posición relativa.
- Rangos genéricos, que representan conjuntos de ubicaciones, como una trayectoria de un objeto en movimiento genérico.

Las trayectorias semánticas enriquecen la representación de ubicaciones cuando se incluyen anotaciones como actividades y medios de transportes. Una definición más reciente de trayectorias simbólicas corresponde a la representación de trayectorias discretas como una función dependiente del tiempo que toma valores en un dominio categórico, por ejemplo, etiquetas que representan alguna anotación. Una trayectoria simbólica es entonces una secuencia de tuplas de la forma $< (i_1, l_1), ..., (i_n, l_n) >$ donde $i_j = [t_j^s, t_j^e]$ es un intervalo de tiempo y l_j es una etiqueta.

Pese a los esfuerzos por construir un modelo de datos para objetos en movimiento que reflejen los casos de la vida real, estos trabajos no han considerado el hecho de que las diversas fuentes de datos no son uniformes en su representación.

Por lo anterior, es necesario construir un modelo de datos para objetos en movimiento que sea capaz de tratar de forma integrada el cómo los objetos se mueven en la actualidad, considerando que las fuentes y las estructuras de representación pueden variar.

3. Modelo integrado para objetos en movimiento

En este capítulo se detalla la solución propuesta que cumple con el objetivo de este trabajo, construir un modelo que soporte distintas representaciones, usando abstracciones de las ubicaciones en tiempo y espacio.

Teniendo en cuenta los modelos existentes, movimientos en espacio libre y movimientos restringido a redes, se prosigue a generalizar y crear un modelo abstracto que soporte cualquier representación de datos, compatibilizando estas representaciones, transformando solo cuando sea necesario.

Se puede decir que de forma abstracta una trayectoria es una secuencia temporal ordenada de referencias espaciales, estas pueden ser coordenadas o símbolos, por ejemplo, la representación de paraderos, edificios, etc. y pueden referirse al espacio libre o embebido en redes.

Los principios básicos que debiese seguir el modelo son:

- Proveer una vista única para usuarios que quieran consultar datos desde distintos modelos.
- Integrar datos de distintas representaciones usando funciones de mapeo entre dominios.
- Definir la tolerancia que se necesita en el proceso de mapeo para manejar a la imprecisión inherente de los datos de localización espacial y temporal. Por ejemplo, la localización puntual de un paradero no es necesariamente la misma posición de un bus en ese paradero.
- Definir relaciones entre trayectoria para permitir la comparación de trayectorias en la definición de restricciones del modelo.

A partir del avance en la definición del modelo de integración de datos, se obtienen las definiciones relevantes para este trabajo.

Elementos básicos del modelo

Definición 1 (Estructura espacial) Una estructura espacial es de la forma $\mathfrak{P} = (Dom_{\mathfrak{P}}, SIden_{\mathfrak{P}}, IdToDom_{\mathfrak{P}})$, donde $\mathsf{Dom}_{\mathfrak{P}}$ es un conjunto infinito de elementos espaciales, llamado dominio espacial; $\mathsf{SIden}_{\mathfrak{P}}$ es un conjunto de identificadores e $\mathsf{IdToDom}_{\mathfrak{P}}$ es una función de mapeo desde los identificadores a elementos del espacio $\mathsf{IdToDom}_{\mathfrak{P}} : \mathsf{SIden}_{\mathfrak{P}} \to 2^{\mathsf{Dom}\mathfrak{P}}$, dejando la posibilidad de que un identificador pueda mapear a múltiples elementos del dominio.

Esta definición es lo suficientemente general para dar cuenta de los diferentes tipos de espacios y representaciones. En efecto, el espacio puede ser un conjunto de puntos, un conjunto de celdas o conjuntos de nodos y arcos que representan una red.

En forma explícita con identificadores que se distinguen de los elementos del dominio es posible diferenciar la referencia al espacio a través de, por ejemplo, el nombre de los lugares, de la representación espacial. Así podemos referirnos a una subdivisión política del espacio por su nombre o referirnos a una paradero en una red de transporte por un etiqueta. Se debe considerar, sin embargo, que la función de mapeo también puede ser la identidad, donde los identificadores también son los elementos del espacio.

De manera similar, se puede definir una estructura temporal.

Definición 2 (Estructura temporal) Una estructura temporal es de la forma $\mathfrak{T} = (\mathsf{Dom}_{\mathfrak{T}}, \mathsf{TIden}_{\mathfrak{T}}, \mathsf{IdToDom}_{\mathfrak{T}})$, donde $\mathsf{Dom}_{\mathfrak{T}}$ es un conjunto infinito de elementos temporales, llamado dominio temporal; $\mathsf{TIden}_{\mathfrak{T}}$ es un conjunto de identificadores e $\mathsf{IdToDom}_{\mathfrak{T}}$ es una función de mapeo desde los identificadores a elementos del espacio $\mathsf{IdToDom}_{\mathfrak{T}} : \mathsf{TIden}_{\mathfrak{T}} \to 2^{\mathsf{Dom}_{\mathfrak{T}}}$.

Definición 3 (Operadores sobre Estructuras) A continuación se definen operaciones básicas relevantes para comparar y transformar representaciones.

Sea $\mathfrak{G}$ una estructura espacial o temporal, los operadores utilizados son:

1. $\mathsf{Dist} : 2^{\mathsf{Dom}_{\mathfrak{G}}} \times 2^{\mathsf{Dom}_{\mathfrak{G}}} \to \mathsf{Int}$. Para una mayor especificación de operadores y restricciones, es útil proveer una función de distancia para comparar elementos en el dominio de una estructura. Dos elementos se consideran equivalentes si tienen una distancia 0.

   El operador de distancia puede usar una tolerancia, la cual es útil en el contexto de representaciones múltiples. Este valor de tolerancia depende también de la representación del dominio y su definición puede hacer uso de la función de distancia en cada dominio. Nos referimos por $\Delta_{\mathfrak{G}}$ al umbral o tolerancia para el dominio en la estructura $\mathfrak{G}$.

2. $\mathsf{Map} : 2^{\mathsf{Dom}_{\mathfrak{G}}} \to 2^{\mathsf{Dom}_{\mathfrak{G}'}}$. Este operador mapea elementos desde un dominio a otro. Este es un operador básico que permite integrar datos de diferentes fuentes al asignarlos a un dominio común de referencia, el cual depende de la aplicación.

Definición 4 (Trayectoria) Contando con definiciones de espacio y tiempo, se introduce el concepto de trayectoria como una combinación de ambos.

Sea $\mathfrak{P}$ una estructura espacial y $\mathfrak{T}$ una estructura temporal, un camino es una

secuencia $< s_1, s_2, ..., s_n >\in \mathsf{SIden}^n$, luego una trayectoria es una secuencia $< (s_1, t_1), (s_2, t_2), ..., (s_n, t_n) >\in (\mathsf{SIden} \times \mathsf{TIden})^n$ tal que para todo $0 < i < j \leq n, t_i < t_j$.

Esta definición de trayectoria se puede complementar con atributos temáticos adicionales, como el modo de transporte, el recorrido, entre otros. Para lograr esto se puede añadir a la definición de trayectoria una estructura nueva, que contenga los atributos temáticos adicionales, por ejemplo, esta nueva estructura podría ser una secuencia de elementos dentro de un conjunto de medios de transporte, de tal forma que una trayectoria podría ser definida como $< (s_1, t_1, tem_1), ..., (s_n, t_n, tem_n) >$, donde t_i son elementos de la estructura temática.

Definición 5 (Operadores básicos sobre trayectorias) Los siguientes operadores constituyen un mínimo para operar y realizar consultas sobre trayectorias en este modelo.

Sea $\mathsf{Traj}_{\langle \mathfrak{P}, \mathfrak{T} \rangle}$ un conjunto de trayectorias definidas sobre las estructuras $\mathfrak{P} = (Dom_{\mathfrak{P}}, \mathsf{SIden}_{\mathfrak{P}}, \mathsf{IdToDom}_{\mathfrak{P}})$ y $\mathfrak{T} = (Dom_{\mathfrak{T}}, \mathsf{TIden}_{\mathfrak{T}}, \mathsf{IdToDom}_{\mathfrak{T}})$, algunos operadores útiles son:

1. $\mathsf{getPath}_{\langle \mathfrak{P}, \mathfrak{T} \rangle}$:$\mathsf{Traj}_{\langle \mathfrak{P}, \mathfrak{T} \rangle} \rightarrow \mathsf{Path}_{\langle \mathfrak{P}, \mathfrak{T} \rangle}$. Este operador retorna la proyección espacial de una trayectoria.
2. $\mathsf{getInterval}_{\langle \mathfrak{P}, \mathfrak{T} \rangle}$:$\mathsf{Traj}_{\langle \mathfrak{P}, \mathfrak{T} \rangle} \rightarrow [\mathsf{TIden}, \mathsf{TIden}]$. Este operador retorna las referencias temporales de la posición inicial y final de una trayectoria.
3. $\mathsf{spatialRange}_{\langle \mathfrak{P}, \mathfrak{T} \rangle}$:$\mathsf{Traj}_{\langle \mathfrak{P}, \mathfrak{T} \rangle} \times (Dom_{\mathfrak{P}})^2 \rightarrow$ Boolean. Este operador retorna Verdad si en algún instante de tiempo la trayectoria estuvo dentro de la región definida por dos ubicaciones en el dominio espacial correspondiente; en el caso del espacio Euclidiano, una región es expresada por puntos extremos opuestos formando un cuadrado, en el caso de espacio en redes, una porción de red puede ser definida usando dos puntos de esta y así obtener el camino más corto entre estos puntos.
4. $\mathsf{temporalRange}_{\langle \mathfrak{P}, \mathfrak{T} \rangle}$:$\mathsf{Traj}_{\langle \mathfrak{P}, \mathfrak{T} \rangle} \times (Dom_{\mathfrak{T}})^2 \rightarrow$ Boolean. Este operador retorna Verdad si la trayectoria existió dentro del intervalo definido por los dos identificadores temporales.

El siguiente ejemplo muestra cómo se aplica esta definición de trayectoria a un caso real con distintas fuentes de datos.

Ejemplo 1 Sea $\mathsf{Traj}_{\langle \mathsf{MT}, \mathfrak{T} \rangle}$ un conjunto de trayectorias que representan los viajes de los usuarios extraídos desde los registros de uso de las smartcards en el sistema de transporte público. Estos viajes son secuencias de tuplas (s, t), donde s es un segmento del camino compuesto por un paradero de subida y uno de bajada de un

usuario en cierto servicio, t el intervalo de tiempo en el que el segmento existió, en este caso, la estructura MT tiene una función de mapeo que toma el nombre del paradero y lo asocia a una ubicación geográfica. La estructura $\mathfrak{T}$ define intervalos usando la función identidad como función de mapeo entre las referencias y el dominio. En el mismo ejemplo, también se tiene el conjunto de trayectorias $\mathsf{Traj}_{\langle \mathsf{MP},\mathfrak{T}\rangle}$ que representan el movimiento de buses o servicios del transporte público. Los datos son capturados mediante sensores GPS ubicados en cada bus y las trayectorias están compuestas de la forma (s', t'), donde MP es una estructura espacial con segmentos s' del dominio espacial, con la función identidad como función de mapeo.

□

La especificación de una trayectoria como una secuencia de tuplas (s,t) es una abstracción simple, que en la práctica puede requerir distintas estructuras como se mostrará en la implementación de este modelo. Sin embargo, se toma esta abstracción para simplificar el concepto de las relaciones entre trayectorias, útil para modelar las restricciones semánticas.

Al comparar con el trabajo en [7], este trabajo se enfoca en modelos discretos para objetos en movimiento, pero se utiliza un modelo abstracto que permita una visión integrada de múltiples modelos discretos.

Definición 6 (Base de datos de objetos en movimiento) Extendiendo la definición clásica de una base de datos relacional, un *esquema de base de datos* para la integración de datos de objetos en movimiento, es de la forma $\Sigma = (\mathcal{U}, \mathcal{S}, \mathcal{T}, \mathcal{A}, \mathcal{R})$, donde: (a) $\mathcal{U}$ es el dominio posiblemente infinito de valores temáticos atómicos. (b) $\mathcal{S}$ un conjunto de estructuras espaciales. (c) $\mathcal{T}$ un conjunto de estructuras temporales. (d) $\mathcal{A}$ es un conjunto de atributos cuyos valores son valores temáticos de $\mathcal{U}$ o trayectorias en $\mathsf{Traj}_{\langle \mathfrak{P},\mathfrak{T}\rangle}$, con $\mathfrak{P} \in \mathcal{S}$ y $\mathfrak{T} \in \mathcal{T}$. (e) $\mathcal{R}$ es un conjunto finito de predicados cuyos atributos pertenecen a $\mathcal{A}$.

Una instancia de base de datos D de un esquema de objetos en movimiento Σ está compuesta de instancias de predicados, los cuales son colecciones finitas de tuplas de la forma $R(c_1, ..., c_n, tr)$[1], donde $R \in \mathcal{R}$, $\langle c_1, ..., c_n\rangle \in \mathcal{U}^n$ contiene los valores de atributos temáticos y $tr \in \mathsf{Traj}_{\langle \mathfrak{P},\mathfrak{T}\rangle}$, considerando $\mathfrak{P} \in \mathcal{S}$ y $\mathfrak{T} \in \mathcal{T}$, es una trayectoria. El conjunto de todas las tuplas de R en Σ se denota como $\mathsf{Tuples}_{\langle \Sigma,R\rangle}$.

Un esquema de base de datos para la integración de datos con varias estructuras espaciales y temporales permite considerar distintos modelos discretos para representar objetos en movimiento, que es de particular interés para este trabajo. Por lo tanto, una instancia puede tener varios predicados y cada uno de ellos con sus

[1]Por simplicidad y por razones prácticas, se considera que tuplas con solo un atributo como trayectoria, pueden ser generalizadas como múltiples trayectorias por tupla

estructuras espaciales y temporales.

Ejemplo 2 (continuación de Ejemplo 1) Formalizando la base de datos del sistema de transporte, se tiene un esquema $\Sigma = (\mathcal{U}, \mathcal{S}, \mathcal{T}, \mathcal{A}, \mathcal{R})$, donde $\mathcal{R}$ es un conjunto que contiene los siguientes predicados $\mathsf{GPS}(\texttt{Plate}, \texttt{Service}, \mathsf{Seq}, \mathsf{T}_g)$ y $\texttt{GTFS}(\texttt{Service}, \mathsf{Seq}, \mathsf{T}_f)$. El predicado GPS almacena las trayectorias (atributo T_g) de buses (identificados por atributo `Plate`) de un servicio (`Service`) durante un día (Seq) según lo registrado en los sensores GPS. `GTFS` es la trayectoria (atributo T_f) de servicios (identificados por el atributo `Service`) durante un día (Seq), definido y entregado por la entidad principal del sistema de transporte público. En esta base de datos: $\mathcal{U}$ es un conjunto con todos los posibles atributos temáticos `Plate` y `Service`. $\mathcal{S}$ es un conjunto de estructuras espaciales MP y MT que representan segmentos espaciales definidos por coordenadas en el espacio libre y un segmento espacial definido por paraderos, respectivamente. $\mathcal{T}$ es el conjunto de una estructura temporal única $\mathfrak{T}$ con intervalos de tiempo como elemento principal. Finalmente, T_g y T_f son atributos de trayectorias cuyos valores están en $\mathsf{Traj}_{\langle \mathsf{MP}, \mathfrak{T} \rangle}$ y $\mathsf{Traj}_{\langle \mathsf{MT}, \mathfrak{T} \rangle}$.

□

4. Implementación

Esta implementación fue realizada en Secondo [13], el cual es un ambiente ideal para crear modelos de datos y sus implementaciones, ya que estos pueden integrarse al sistema de gestión de base de datos a través de módulos llamados álgebras. Un álgebra en Secondo se compone de tipos de datos y operadores sobre estos y se utiliza usualmente cuando los investigadores necesitan comparar sus trabajos, ya sean modelos, índices o estructuras frente a modelos del estado del arte. En este trabajo fueron utilizadas dos álgebras existentes en Secondo, que definen tipos de objetos en movimiento : (1) Temporal Algebra para objetos en movimiento libre, con un tipo de dato llamado **MPoint** (MP) [10] y (2) Network Algebra para movimiento en redes, con un tipo de dato llamado **MGPoint** (MG) [12]. Adicionalmente se crea otra álgebra (3) Transportation Algebra, con un tipo de dato llamado **MTPoint**(MT).

Tomando en consideración la base teórica de este trabajo y los datos disponibles del sistema de transporte, se presentan tres estructuras espaciales y una estructura temporal que buscan representar un modelo de base de datos que cubra las necesidades de un sistema de transporte.

Es importante señalar que en esta implementación no se hace uso de optimización de accesos en base a mecanismos de indexación, ya que se encuentra fuera del alcance de este trabajo y se plantea como trabajo futuro.

4.1. Estructuras

En MPoint las trayectorias son modeladas como secuencias de intervalos de tiempo asociados a pares de coordenadas (segmentos espaciales) en el espacio libre. En términos de la definición de este trabajo, este modelo considera una estructura temporal $\mathfrak{T} = (Dom_{\mathfrak{T}}, \mathsf{TIden}_{\mathfrak{T}}, \mathsf{IdToDom}_{\mathfrak{T}})$, donde $Dom_{\mathfrak{T}}$ es un conjunto infinito de intervalos de tiempo abiertos en su lado derecho, de la forma $[\cdot, \cdot)$, $\mathsf{TIden}_{\mathfrak{T}}$ es un subconjunto de intervalos en $Dom_{\mathfrak{T}}$ y $\mathsf{IdToDom}_{\mathfrak{T}}$ la función identidad. Además, se considera la estructura espacial $\mathfrak{P}_{\mathsf{MP}} = (Dom_{\mathfrak{P}_{\mathsf{MP}}}, \mathsf{SIden}_{\mathfrak{P}_{\mathsf{MP}}}, \mathsf{IdToDom}_{\mathfrak{P}_{\mathsf{MP}}})$, donde $Dom_{\mathfrak{P}_{\mathsf{MP}}}$ es un conjunto infinito de coordenadas y segmentos espaciales, $\mathsf{SIden}_{\mathfrak{P}_{\mathsf{MP}}}$ es el subconjunto de segmentos espaciales (lineas) y $\mathsf{IdToDom}_{\mathfrak{P}_{\mathsf{MP}}}$ es la función identidad.

En MGPoint las trayectorias son modeladas como secuencias de intervalos de tiempo asociados a un par de posiciones (segmentos de red) sobre redes. Este modelo considera una estructura espacial $\mathfrak{P}_{\mathsf{MG}} = (Dom_{\mathfrak{P}_{\mathsf{MG}}}, \mathsf{SIden}_{\mathfrak{P}_{\mathsf{MG}}}, \mathsf{IdToDom}_{\mathfrak{P}_{\mathsf{MG}}})$, donde $Dom_{\mathfrak{P}_{\mathsf{MG}}}$ es un conjunto infinito de posiciones y segmentos sobre redes, $\mathsf{SIden}_{\mathfrak{P}_{\mathsf{MG}}}$ es un conjunto de segmentos de red, $\mathsf{IdToDom}_{\mathfrak{P}_{\mathsf{MG}}}$ es la función identidad.

Como se mencionó anteriormente, se creó una nueva álgebra llamada Transportation Algebra con el tipo de dato MTPoint (MT). Este modela situaciones donde las trayectorias son representadas como segmentos espaciales definidos por pares de referencias espaciales tales como paraderos del transporte público. En este caso la estructura espacial corresponde a $\mathfrak{P}_{\mathsf{MT}} = (Dom_{\mathfrak{P}_{\mathsf{MT}}}, \mathsf{SIden}_{\mathfrak{P}_{\mathsf{MT}}}, \mathsf{IdToDom}_{\mathfrak{P}_{\mathsf{MT}}})$, donde $Dom_{\mathfrak{P}_{\mathsf{MT}}}$ es un conjunto infinito de coordenadas y segmentos espaciales, $\mathsf{SIden}_{\mathfrak{P}_{\mathsf{MT}}}$ es un conjunto de tuplas de la forma $(\mathsf{ST}_1, \mathsf{ST}_2)$, siendo ST_1 la referencia al paradero de partida y ST_2 al paradero de llegada. Tiene una función de mapeo $\mathsf{IdToDom}_{\mathfrak{P}_{\mathsf{MT}}}$ que toma referencias espaciales (paraderos) y las mapea a tuplas de coordenadas en $Dom_{\mathfrak{T}_{\mathsf{MT}}}$.

Finalmente, se implementa un álgebra general para objetos en movimiento para la integración de distintas representaciones. Esta álgebra define un objeto en movimiento general $\mathsf{GMObject}$ (GM), donde las trayectorias son secuencias de intervalos de tiempo asociadas con segmentos espaciales definidos con una estructura espacial $\mathfrak{P}_{\mathsf{GM}} = (Dom_{\mathfrak{P}_{\mathsf{GM}}}, \mathsf{SIden}_{\mathfrak{P}_{\mathsf{GM}}}, \mathsf{IdToDom}_{\mathfrak{P}_{\mathsf{GM}}})$, donde $Dom_{\mathfrak{P}_{\mathsf{GM}}} = Dom_{\mathfrak{P}_{\mathsf{MP}}} \cup Dom_{\mathfrak{P}_{\mathsf{MG}}} \cup Dom_{\mathfrak{P}_{\mathsf{MT}}}$, $\mathsf{TIden}_{\mathfrak{P}_{\mathsf{GM}}} = \mathsf{SIden}_{\mathfrak{P}_{\mathsf{MP}}} \cup \mathsf{SIden}_{\mathfrak{P}_{\mathsf{MG}}} \cup \mathsf{SIden}_{\mathfrak{P}_{\mathsf{MT}}}$, luego por consecuencia $\mathsf{IdToDom}_{\mathfrak{P}_{\mathsf{GM}}} = \mathsf{IdToDom}_{\mathfrak{P}_{\mathsf{MP}}} \cup \mathsf{IdToDom}_{\mathfrak{P}_{\mathsf{MG}}} \cup \mathsf{IdToDom}_{\mathfrak{P}_{\mathsf{MT}}}$.

El propósito principal de esta álgebra genérica es preservar los datos en su representación original, permitiendo que los datos solo cambien cuando es necesario (usando funciones de mapeo). Esta álgebra incluye diferentes funciones de mapeo, las cuales son clasificadas en dos grupos: (1) las funciones de mapeo desde las referencias espaciales a los elementos en el dominio (e.g., $\mathsf{IdToDom}_{\mathfrak{P}_{MT}}$). Sólo $MTPoint$ tiene una función de mapeo distinta a la identidad, ya que los identificadores de paraderos son mapeados a coordenadas en el dominio de espacio libre. $\mathfrak{P}_{\mathsf{MT}}$ y $\mathfrak{P}_{\mathsf{MG}}$ utilizan la función identidad. (2) Las funciones de mapeo desde un dominio espacial a otro, la cual se menciona en Definición (1) usando notación $\mathsf{Map}_{\langle \mathfrak{G}, \mathfrak{G}' \rangle}$ y es instanciada en distintas implementaciones para distintas estructuras espaciales como se describe en el Cuadro 1.

Operador	Sintaxis
MPointToMGPoint	$\mathsf{Traj}_{\langle \mathfrak{P}_{\mathsf{MP}}, \mathfrak{T}_{\mathsf{MP}} \rangle} \rightarrow \mathsf{Traj}_{\langle \mathfrak{P}_{\mathsf{MG}}, \mathfrak{T}_{\mathsf{MG}} \rangle}$
MGPointToMPoint	$\mathsf{Traj}_{\langle \mathfrak{P}_{\mathsf{MG}}, \mathfrak{T}_{\mathsf{MG}} \rangle} \rightarrow \mathsf{Traj}_{\langle \mathfrak{P}_{\mathsf{MP}}, \mathfrak{T}_{\mathsf{MP}} \rangle}$
MTpointToMPoint	$\mathsf{Traj}_{\langle \mathfrak{P}_{\mathsf{MT}}, \mathfrak{T}_{\mathsf{MT}} \rangle} \rightarrow \mathsf{Traj}_{\langle \mathfrak{P}_{\mathsf{MP}}, \mathfrak{T}_{\mathsf{MP}} \rangle}$
MTpointToMGPoint	$\mathsf{Traj}_{\langle \mathfrak{P}_{\mathsf{MT}}, \mathfrak{T}_{\mathsf{MT}} \rangle} \rightarrow \mathsf{Traj}_{\langle \mathfrak{P}_{\mathsf{MG}}, \mathfrak{T}_{\mathsf{MG}} \rangle}$

Cuadro 1: Operadores para funciones de mapeo

4.1.1. Punto Genérico en Movimiento

Para realizar la implementación de forma general y escalable, se crea un tipo de dato que pueda representar cualquier tipo de trayectoria y pueda ser almacenado en su dominio original, permitiendo realizar transformaciones sólo cuando sea necesario. Este tipo de datos se denomina como GMO, sirve para encapsular los operadores y siempre consultar usando un tipo de dato, el cual internamente discrimina qué tipo de trayectoria necesita procesar.

Este tipo de dato, contiene **N** variables de tipo trayectoria y **N** booleanos, siendo **N** la cantidad de tipos de trayectorias representadas en él, donde solo se almacena la trayectoria que se desea representar, puesto que el resto de las trayectorias pasa a ser vacía. Los **N** booleanos sirven para identificar cuál es la trayectoria activa.

Para el caso práctico de este trabajo, el tipo de dato GMO se establece de la siguiente forma:

```
//Class modeling Generic Moving Point
class GenericMPoint
    boolean mgp, mtp, mp
    mpoint mpoint_p
    mgpoint mgpoint_p
    mtpoint mtpoint_p
```

4.1.2. Funciones de Mapeo

En este modelo existen dos tipos de funciones de mapeo, el primer tipo corresponde a funciones de mapeo a dominios (mencionado en la definición de estructura espacial y temporal), en adelante función de mapeo tipo ϕ. Este tipo de función sirve para que los elementos de una estructura espacial que contienen identificadores, puedan tener una representación en el dominio espacial; por ejemplo, en la estructura MT los identificadores son nombres de paraderos del sistema de transporte, luego se necesita una función que los lleve al dominio del espacio libre.

El segundo tipo de función de mapeo corresponde a mapeos entre estructuras espaciales, en adelante función de mapeo tipo ρ. Esta toma un elemento desde una estructura espacial y lo lleva a otra estructura espacial.

4.1.3. Operadores

En el siguiente listado se detalla la implementación de los operadores definidos en la definición 5 del capítulo 3, se añade además el constructor para el tipo de dato

GMObject.

- creategenericmpoint : [mpoint||*mgpoint*||*mtpoint*]→ genericmpoint
- temporal_range : genericmpoint x [instant||*periods*]→ boolean
- get_interval : genericmpoint → Periods
- spatial_range : genericmpoint x genericpoint x genericpoint → boolean

El operador **creategenericmpoint** crea un GMO, punto genérico en movimiento que puede contener cualquier tipo de trayectoria mencionada anteriormente.

Los operadores **temporal_range** y **get_interval**, son operadores temporales sobre GMO. El primero interseca el intervalo de tiempo de la trayectoria con un instante o periodo, mientras que **get_interval** entrega el o los periodos donde la trayectoria existe.

Para el tipo de dato GMO se crea el operador **spatial_range** que sirve para consultar intersecciones con rangos espaciales, retornado True si existe tal intersección. Este toma distinto significado de acuerdo a la estructura espacial que lo utilice. Por ejemplo, para MP, este operador toma dos identificadores espaciales y crea un bounding box para intersecarlo con la trayectoria. Para MG no existe la noción de bounding box, por lo cual el operador toma dos identificadores y crea el camino más corto en la red entre estos para intersecarlo con la trayectoria. Finalmente para MT, toma ambos identificadores y si en algún punto la trayectoria interseca a alguno de los identificadores la respuesta es True.

5. Caso de Estudio y Evaluación: Esquema de un sistema de transporte

Para el caso de estudio de este trabajo se obtuvieron datos del sistema de transporte público de Santiago de Chile, estos datos provienen de distintas fuentes: (1) GTFS (General Transit Feed Specification), entregados por la DTPM (Dirección de Transporte Público Metropolitano) que representan la oferta del sistema de transporte, aquí se encuentra el cronograma de los Servicios. (2) Registros Bip!, entregados por la DTPM, reflejan el uso del sistema de transporte, se encuentran los registros de viajes de los usuarios que utilizan smartcards. (3) Registros GPS, entregados por la DTPM, reflejan la oferta real del sistema de transporte público, ya que son los registros de los dispositivos GPS instalados en cada uno de los buses.

A continuación, se muestra el formato de los datos originales para las fuentes GTFS, tarjeta inteligente *Bip!* y GPS. Es importante mencionar que las distintas fuentes presentan diferencias en la codificación y en el formato de los datos.

idRuta	idAgencia	nombreCortoRuta	nombreLargoRuta	tipoRuta
101	*TS*	*101*	*Recoleta - Cerrillos*	*3*
101c	*TS*	*101c*	*(M) Blanqueado - Cerrillos*	*3*
102	*TS*	*102*	*(M) Blanqueado - Mall Plaza Tobalaba*	*3*

Cuadro 2: Fuente de datos GTFS - Tabla Rutas.

idViaje	idRuta	viajeDirecto	idDireccion	idServicio
101-I-L_ V23-B00	*101*	*Cerrillos*	*0*	*L_ V23*
101-I-L_ V23-B02	*101*	*Cerrillos*	*0*	*L_ V23*
101-I-L_ V23-B03	*101*	*Cerrillos*	*0*	*L_ V23*

Cuadro 3: Fuente de datos GTFS - Tabla Viajes.

idParadero	codParadero	nombreParadero	latParadero	longParadero
PB1	*PB1*	*PB1-Venezuela Esq. / Bolivia*	*-33.4045537555341*	*-70.623095148163*
PB2	*PB2*	*PB2-Venezuela Esq. / H. De La Concepción*	*-33.402453078379*	*-70.6266392477005*
PB3	*PB3*	*PB3-Reina De Chile Esq. / Avenida El Salto*	*-33.4012186446509*	*-70.6297346535453*

Cuadro 4: Fuente de datos GTFS - Tabla Paraderos.

patente	latitud	longitud	TiempoGPS	Ruta	Velocidad	Ignición
BBJZ-76	*-33.5986404*	*-70.5814056*	*20160523172009*	*F07 00R*	*37.97*	*SI*
BBJZ-76	*-33.5981827*	*-70.5788727*	*20160523172039*	*F07 00R*	*0*	*SI*
BBJZ-76	*-33.5981827*	*-70.5788727*	*20160523172109*	*F07 00R*	*0*	*SI*

Cuadro 9: Fuente de datos GPS de buses.

id	idViaje	tLlegada	tPartida	idParadero	secuenciaParadas
1	*101-I-L_ V23-B00*	*00:00:00*	*00:00:00*	*PB1*	*1*
2	*101-I-L_ V23-B00*	*00:00:59*	*00:00:59*	*PB2*	*2*
3	*101-I-L_ V23-B00*	*00:01:46*	*00:01:46*	*PB3*	*3*

Cuadro 5: Fuente de datos GTFS - Tabla Tiempos de paradas.

idServicio	lunes	martes	miercoles	jueves	viernes	sabado	domingo	fechaInicio	fechaTermino
L_ V23	*1*	*1*	*1*	*1*	*1*	*0*	*0*	*20160430*	*20160710*
S_ V23	*0*	*0*	*0*	*0*	*0*	*1*	*0*	*20160430*	*20160710*
D_ V23	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*20160430*	*20160710*

Cuadro 6: Fuente de datos GTFS - Tabla Calendario (Servicios).

idUsuario	nViaje	nEtapas	paraderoSubida	paraderoBajada	tSubida	tBajada
1091258	*1*	*1*	*L-34-2-60-OP*	*L-34-76-20-SN*	*2016-05-26 15:58:29*	*2016-05-26 16:15:42*
1091258	*2*	*1*	*L-34-76-15-SN*	*L-34-2-45-PO*	*2016-05-26 18:00:35*	*2016-05-26 18:34:09*
1091258	*3*	*1*	*L-34-2-50-OP*	-	*2016-05-27 12:10:55*	-
1091258	*4*	*2*	*Plaza de Puente Alto*	-	*2016-05-29 13:56:28*	-

Cuadro 7: Fuente de datos Tarjeta inteligente *Bip!* - Tabla Viajes.

idUsuario	tipoTransporte	servicioSentidoVar	nViaje	nEtapa	paraderoSubida	paraderoBajada	tSubida	tBajada
1091258	*Bus*	*F01 00R*	*1*	*1*	*L-34-2-60-OP*	*L-34-76-20-SN*	*2016-05-26 15:58:29*	*2016-05-26 16:15:42*
1091258	*Bus*	*F17 00I*	*2*	*1*	*L-34-76-15-SN*	*L-34-2-45-PO*	*2016-05-26 18:00:35*	*2016-05-26 18:34:09*
1091258	*Bus*	*F01 00R*	*3*	*1*	*L-34-2-50-OP*	-	*2016-05-27 12:10:55*	-
1091258	*Metro*	*L4*	*4*	*1*	*Plaza de Puente Alto*	*Los Dominicos*	*2016-05-29 13:56:28*	*2016-05-29 14:44:45*
1091258	*Bus*	*B52 00R*	*4*	*2*	*T-17-141-SN-2*	-	*2016-05-29 14:59:14*	-

Cuadro 8: Fuente de datos Tarjeta inteligente *Bip!* - Tabla Etapas.

Los datos recopilados pertenecen al día 23-06-2016, el tamaño de los datos son 737MB para GPS, 53MB para GTFS y 8.2 GB para Bip!.

5.1. Bases de datos de Transporte público

En base a las álgebra usadas en Secondo, se define una base de datos que integra los datos provenientes de las distintas fuentes de datos del sistema de transporte público.

Esta base de datos contempla los siguientes esquema de relaciones: $MPoint$, el cual representa la información provista por los buses del transporte público de Santiago, $MTPoint$ representa la información provista por la programación de los buses (GTFS), vale decir, las trayectorias ideales que los buses debieran cumplir. $MGPoint$ representa, al igual que $MPoint$, la información real provista por los buses, obtenidos mediante dispositivos GPS.

5.1.1. Relaciones

El esquema que modela los datos del sistema de transporte público, considerando los tipos de datos de objetos en movimiento (MPoint, MGPoint, MTPoint), es el siguiente:

- GPS_MPoint(LicensePlate: int, Service: str, Sequence: int, Journey: MPoint)
- GPS_MGPoint(LicensePlate: str, Service: str, Sequence: int, Journey: MGPoint)
- GTFS(Service: str, Sequence: int, Journey: MTPoint)
- User(Id : int, Sequence: int, Journey: MTPoint, Services : List)

Por otro lado, es necesario para efectos de la implementación tener el mismo esquema, pero con las trayectorias almacenadas con el tipo de dato GMO, así la integración de datos puede ser llevada a cabo de una forma más clara. En consecuencia, el esquema es el siguiente:

- GPS_MPoint(LicensePlate: int, Service: str, Sequence: int, Journey: GMO)
- GPS_MGPoint(LicensePlate: int, Service: str, Sequence: int, Journey: GMO)
- GTFS(Service: str, Sequence: int, Journey: MTPoint)
- User(Id : int, Sequence: int, Journey: GMO, ThematicJourney : List)

Se evalúa este modelo de dos formas, primero de forma experimental, haciendo uso de los datos del transporte público de Santiago y, en segundo lugar, la capacidad de expresividad. Esto quiere decir, que el modelo con los tipos de datos y operadores es capaz de responder a las mismas consultas que los modelos de objetos en movimiento mencionados en capítulos anteriores.

5.2. Consultas

Para evaluar este modelo se procede a realizar consultas espaciales y temporales. Las consultas son creadas para cada tipo de trayectoria y también para el tipo de dato GMO con el fin de demostrar que usando este modelo y su tipo de dato principal, se puede realizar el mismo conjunto de consultas.

Se consideran 2 variantes para cada consulta, una con un rango L (en adelante tipo 1) y otra con un rango S (en adelante tipo 2).

1. **Space_L** como las consultas espaciales sobre el mayor rango posible y **Time_L** como las consultas temporales sobre el mayor rango de tiempo de los datos (1 día entero).

2. **Space_S** y **Time_S** que corresponden a fracciones pequeñas (30 %) de la primera variante.

Las consultas se categorizan en:

- Consultas temporales por rango
- Consultas espaciales por rango (dado dos identificadores espaciales)

Las consultas, escritas en Secondo y SQL, son las siguientes (usando consultas tipo 1 para ejemplificar):

- Temporal
  - MPoint
    query GPS_MPoint **feed extend**[Present : .Journey **present**(time_L)]
    Select MPoint.*, **present**(MPoint.Journey, time_L) as present
    from GPS_MPoint as MPoint
  - MGPoint
    query GPS_MGPoint **feed extend**[Present : .Journey **present**(time_L)]
    Select MGPoint.*, **present**(MGPoint.Journey, time_L) as present
    from GPS_MGPoint as MGPoint
  - MTPoint
    query GTFS **feed extend**[Present : .Journey mt_**present**(time_L)]
    Select GTFS.*, **present**(GTFS.Journey, time_L) as present
    from GTFS
  - GenericMPoint
    query GenericMPoint_Table **feed extend**[Present : **temporal_range**(.Journey, time_L)]
    Select GMP.*, **temporal_range**(GMP.Journey, time_L, time_L)
    from GenericMPoint_Table as GMP
- Espacial
  - MPoint
    query GPS_MPoint **feed extend**[Between : trajectory(.Journey) **inside** region_L]
    Select MPoint.*, **inside**(**trajectory**(MPoint.Journey), region_L)
    from GPS_MPoint as MPoint
  - MGPoint
    query GPS_MGPoint **feed extend**[Between : .Journey **passes shortest_path**(start_L, end_L)]
    Select MGPoint.*, **passes**(MGPoint.Journey, **shortest_path**(start_L,

end_L)
from GPS_MGPoint as MGPoint

- MTPoint
  query GTFS **feed extend**[Between : **mt_between**(.Journey, start_L, end_L)]
  Select GTFS.*, **passes**(GTFS.Journey, **shortest_path**(start_L, end_L)
  from GTFS

- GenericMPoint
  query GenericMPoint_Table **feed**
  extend[Between : **spatial_range**(.Journey, genericpoint_start_L, genericpoint_end_L)]
  Select GMP.*, **spatial_range**(GMP.Journey, genericpoint_start_L, genericpoint_end_L)
  from GenericMPoint_Table as GMP

En lo que respecta a consultas por rango, el tipo de dato propuesto GMO puede simular todas las consultas de los otros modelos a través de un sólo operador, tanto para las consultas temporales como espaciales.

Si bien sólo se abordan 3 tipos de trayectorias, el tipo de dato GMO es capaz de añadir cualquier tipo de representación de trayectoria que cumpla con las definiciones de este trabajo, esto quiere decir, que su componente espacial y temporal se pueden mapear a una estructura espacial y temporal.

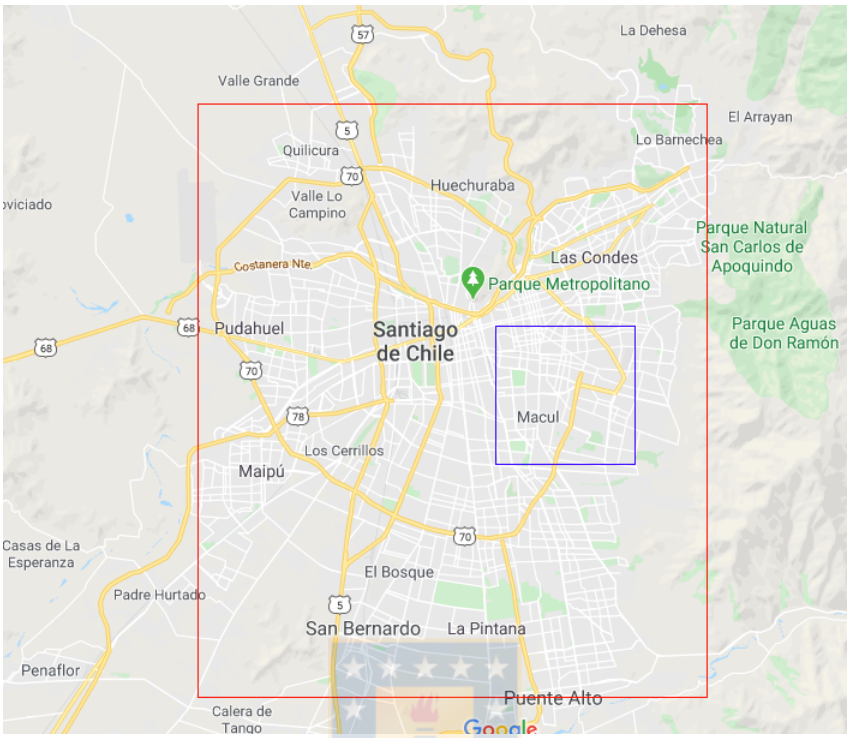

Figura 4: Ejemplo consulta Space en espacio libre, en rojo tamaño L, en azul tamaño S. Fuente: Elaboración Propia

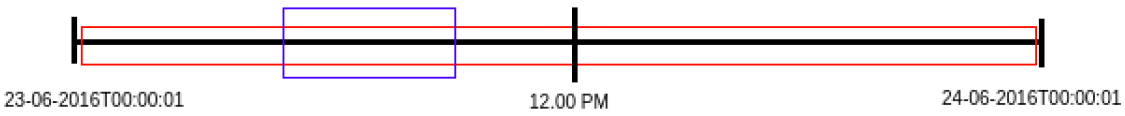

Figura 5: Ejemplo consultas Time, en rojo tamaño L, en azul tamaño S. Fuente: Elaboración Propia

6. Resultados y Análisis

En este capítulo se detallan los resultados. Estos experimentos fueron realizados en una máquina con procesador Intel Xeon CPU E3-1220 v5 3Ghz x 8, memoria RAM de 16 GB y memoria secundaria de 100 GB, sistema operativo Ubuntu 16.04.

Para las consultas tipo 1 y 2, se utilizaron 5000 tuplas por cada tipo de trayectoria, con 100 repeticiones por cada una de las consultas, excepto en el caso de las consultas espaciales de las trayectorias MG que resultaron costosas en tiempo de ejecución, por lo que se realizaron 10 repeticiones.

6.1. Tiempos de Consultas

Resultados en los siguientes cuadros:

	GMO/MP	GMO/MG	GMO/MT
Temporal	0.87	0.97	0.98
Espacial	0.97	0.99	0.87

Cuadro 10: Relación GMO frente a MP, MG y MG usando consultas tipo 1

	GMO/MP	GMO/MG	GMO/MT
Temporal	0.88	0.96	0.86
Espacial	0.97	0.99	0.90

Cuadro 11: Relación de GMO frente a MP, MG y MG usando consultas tipo 2

En términos generales, existe una leve diferencia entre ejecutar consultas sobre los modelos MP, MG y MT, frente a el tipo de dato propuesto GMO de 5.5 %, mientras que en las de tipo 2, GMO presenta una disminución promedio de 5.1 % en el tiempo de consulta en comparación con los otros modelos.

Esto último es explicable debido a que usando un sólo operador para la consulta por rangos se realizan menos procesos (operaciones de entrada y salida) dentro del motor de base de datos.

Para clarificar esta situación, en una consulta por rango espacial para el tipo de dato MP se ejecutan más de un operador lo que conlleva que el resultado debe pasar a otro operador. Esta consulta en Secondo se expresa como:

query GPS_MPoint **feed**
extend[M: .Journey **intersects** space_L **rect2region**]
filter[.M # TRUE]

Como se observa la entrada *space_L* se ingresa en el operador *rect2region* y el resultado es usado en el operador *intersects*. Mientras que en el nuevo tipo de

dato GMO, los parámetros del operador son ingresados y todo es procesado en el mismo operador como se señala en la sección Consultas.

6.2. Uso de Espacio

A continuación se presenta un cuadro comparativo en términos de uso de espacio, considerando que para cada tipo de trayectoria se tiene 5000 tuplas.

Tipo de Trayectoria	Con GMO	Original
MP	64616 Kb	64756 Kb
MT	15432 Kb	15584 Kb
MG	82716 Kb	82872 Kb

Cuadro 12: Uso de espacio para las distintas representaciones

Se observa que en términos generales la diferencia de uso de espacio entre la utilización de GMO frente a los modelos en crudo, no es significativa, alcanzando una diferencia máxima de 0.98 %, donde GMO usa un mayor espacio. En promedio la diferencia de uso de espacio es de 0.46 %, esto se debe a la utilización de variables extras en el tipo de dato GMO, con los **N** punteros y **N** booleanos, donde **N** son los tipos de trayectorias que se puede representar.

7. Conclusión

El modelo propuesto para objetos en movimiento presenta de forma teórica la posibilidad de tratar información de objetos en movimiento independiente de su representación.

En este trabajo se hizo uso de un caso experimental, el transporte público de Santiago de Chile, el cual presenta 3 tipos de representaciones para objetos en movimiento. De acuerdo al modelo desarrollado de este trabajo, cada tipo de dato de distintas fuentes puede ser llevado a un tipo de trayectoria, esto nos hace definir una estructura espacial y temporal para cada una de ellas.

En lo que respecta a las consultas por rango, que es el enfoque de este trabajo, el modelo propuesto es capaz de expresar las mismas consultas en comparación a usar cada tipo de trayectoria por separado, por lo cual se cumple con lo propuesto en la hipótesis.

Por otro lado, se crea un nuevo tipo de dato, llamado GMO, el cual es capaz de almacenar de forma interna cualquier tipo de trayectoria. Este tipo de dato solo transforma su representación cuando sea necesario, por lo cual, no hay mayor costo de almacenamiento extra, esto toma mayor relevancia ya que no hay modificación de los datos originales, pero esto se contrapone con el costo computacional de realizar las transformaciones cuando se requiera.

8. Trabajo Futuro

Al avanzar en esta tesis surgieron nuevas interrogantes para ser abordadas, las cuales se proponen como trabajo futuro.

El primer punto que surge al terminar esta tesis es que si bien se propone un tipo de trayectoria llamado MT este carece de formalidad. El objetivo de este tipo de trayectoria es tratar de representar la red de paraderos del transporte público, pero sin una definición formal de red. Es por esto que se propone trabajar este concepto y llevarlo a una definición de grafo, lo complejo de este objetivo es que eventualmente todos los paraderos pueden estar conectados, convirtiendo el grafo en uno de tipo completo.

Otro de las interrogantes que surgen con este trabajo es el poder diseñar índices espaciales y temporales de tal forma que puedan adaptarse a cualquier tipo de representación de las trayectorias. Junto a la creación de índices para GMO, también es deseable poder crear índices para el tipo de dato MT.

Dado el origen y el formato de los datos de origen del caso práctico en este trabajo no se explotó el uso de las estructuras temporales, debido a que los datos temporales siempre son representados de la misma forma, por ende queda pendiente la incorporación de nuevas estructuras temporales al tipo de dato GMO en el caso que surjan nuevas representaciones.

Sería posible observar si la incorporación de nuevas representaciones temporales afectan al poder de expresividad y al desempeño de las consultas temporales por rango.

Referencias

[1] Carlo Batini, Maurizio Lenzerini, and Shamkant B. Navathe. A comparative analysis of methodologies for database schema integration. *ACM Comput. Surv.*, 18(4):323–364, 1986.

[2] Tiziana Catarci and Maurizio Lenzerini. Interschema knowledge in cooperative information systems. In *CoopIS*, pages 55–62, 1993.

[3] Philippe Cudré-Mauroux, Eugene Wu, and Samuel Madden. Trajstore: An adaptive storage system for very large trajectory data sets. In *Proceedings of the 26th International Conference on Data Engineering, ICDE 2010, March 1-6, 2010, Long Beach, California, USA*, pages 109–120, 2010.

[4] Victor Teixeira de Almeida and Ralf Hartmut Güting. Indexing the trajectories of moving objects in networks. *GeoInformatica*, 9(1):33–60, 2005.

[5] Christian Düntgen, Thomas Behr, and Ralf Hartmut Güting. Berlinmod: A benchmark for moving object databases. *The VLDB Journal*, page 1335, 2009.

[6] Martin Erwig, Ralf Hartmut Guting, Markus Schneider, and Michalis Vazirgiannis. Spatio-temporal data types: An approach to modeling and querying moving objects in databases. *GeoInformatica*, 3(3):269–296, Sep 1999.

[7] Luca Forlizzi, Ralf Hartmut Güting, Enrico Nardelli, and Markus Schneider. A Data Model and Data Structures for Moving Objects Databases. In *SIGMOD Conference*, pages 319–330. ACM, 2000.

[8] Elias Frentzos. Indexing objects moving on fixed networks. In *Advances in Spatial and Temporal Databases, 8th International Symposium, SSTD 2003, Santorini Island, Greece, July 24-27, 2003, Proceedings*, pages 289–305, 2003.

[9] Ralf Hartmut Güting, Michael H. Böhlen, Martin Erwig, Christian S. Jensen, Nikos A. Lorentzos, Enrico Nardelli, Markus Schneider, and Jose Ramon Rios Viqueira. Spatio-temporal models and languages: An approach based on data types. In *Spatio-Temporal Databases: The CHOROCHRONOS Approach*, pages 117–176, 2003.

[10] Ralf Hartmut Güting, Michael H. Böhlen, Martin Erwig, Christian S. Jensen, Nikos A. Lorentzos, Markus Schneider, and Michalis Vazirgiannis. A foundation for representing and querying moving objects. *ACM Trans. Database Syst.*, 25(1):1–42, 2000.

[11] Ralf Hartmut Güting, Victor Teixeira de Almeida, Dirk Ansorge, Thomas Behr, Zhiming Ding, Thomas Höse, Frank Hoffmann, Markus Spiekermann,

and Ulrich Telle. SECONDO: an extensible DBMS platform for research prototyping and teaching. In *Proceedings of the 21st International Conference on Data Engineering, ICDE 2005, 5-8 April 2005, Tokyo, Japan*, pages 1115–1116, 2005.

[12] Ralf Hartmut Güting, Victor Teixeira de Almeida, and Zhiming Ding. Modeling and querying moving objects in networks. *VLDB J.*, 15(2):165–190, 2006.

[13] Ralf Hartmut Güting and Markus Schneider. *Moving Objects Databases*. Morgan Kaufmann, 2005.

[14] Alon Y. Halevy. Answering queries using views: A survey. *VLDB J.*, 10(4):270–294, 2001.

[15] Richard Hull. Managing semantic heterogeneity in databases: A theoretical perspective. In *Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, May 12-14, 1997, Tucson, Arizona, USA*, pages 51–61, 1997.

[16] Luca Leonardi, Salvatore Orlando, Alessandra Raffaetà, Alessandro Roncato, Claudio Silvestri, Gennady L. Andrienko, and Natalia V. Andrienko. A general framework for trajectory data warehousing and visual OLAP. *GeoInformatica*, 18(2):273–312, 2014.

[17] Witold Litwin, Leo Mark, and Nick Roussopoulos. Interoperability of multiple autonomous databases. *ACM Comput. Surv.*, 22(3):267–293, 1990.

[18] Gerasimos Marketos, Elias Frentzos, Irene Ntoutsi, Nikos Pelekis, Alessandra Raffaetà, and Yannis Theodoridis. Building real-world trajectory warehouses. In *Seventh ACM International Workshop on Data Engineering for Wireless and Mobile Access, Mobide 2008, June 13, 2008, Vancouver, British Columbia, Canada, Proceedings*, pages 8–15, 2008.

[19] Marcela A. Munizaga and Carolina Palma. Estimation of a disaggregate multimodal public transport origin–destination matrix from passive smartcard data from santiago, chile. *Transportation Research Part C: Emerging Technologies*, 24:9 – 18, 2012.

[20] Salvatore Orlando, Renzo Orsini, Alessandra Raffaetà, Alessandro Roncato, and Claudio Silvestri. Trajectory data warehouses: Design and implementation issues. *JCSE*, 1(2):211–232, 2007.

[21] Nikos Pelekis and Yannis Theodoridis. *Mobility Data Management and Exploration*. Springer, 2014.

[22] Dieter Pfoser, Christian S. Jensen, and Yannis Theodoridis. Novel approaches to the indexing of moving object trajectories. In *VLDB 2000, Proceedings of 26th International Conference on Very Large Data Bases, September 10-14, 2000, Cairo, Egypt*, pages 395–406, 2000.

[23] Iulian Sandu Popa, Karine Zeitouni, Vincent Oria, Dominique Barth, and Sandrine Vial. Indexing in-network trajectory flows. *VLDB J.*, 20(5):643–669, 2011.

[24] Reasey Praing and Markus Schneider. A universal abstract model for future movements of moving objects. In *The European Information Society: Leading the Way with Geo-information, Proceedings of the 10th AGILE Conference, Aalborg, Denmark, 8-11 May 2007*, pages 111–120, 2007.

[25] A. Prasad Sistla, Ouri Wolfson, Sam Chamberlain, and Son Dao. Modeling and querying moving objects. In *ICDE*, pages 422–432. IEEE Computer Society, 1997.

[26] Stefano Spaccapietra. Editorial: Spatio-temporal data models and languages. *GeoInformatica*, 5(1):5–9, 2001.

[27] Yufei Tao and Dimitris Papadias. Mv3r-tree: A spatio-temporal access method for timestamp and interval queries. In *VLDB 2001, Proceedings of 27th International Conference on Very Large Data Bases, September 11-14, 2001, Roma, Italy*, pages 431–440, 2001.

[28] Gomer Thomas, Glenn R. Thompson, Chin-Wan Chung, Edward Barkmeyer, Fred Carter, Marjorie Templeton, Stephen Fox, and Berl Hartman. Heterogeneous distributed database systems for production use. *ACM Comput. Surv.*, 22(3):237–266, 1990.

[29] Jeffrey D. Ullman. Information integration using logical views. In *Database Theory - ICDT '97, 6th International Conference, Delphi, Greece, January 8-10, 1997, Proceedings*, pages 19–40, 1997.

[30] Ouri Wolfson, Bo Xu, Sam Chamberlain, and Liqin Jiang. Moving objects databases: Issues and solutions. In *10th International Conference on Scientific and Statistical Database Management, Proceedings, Capri, Italy, July 1-3, 1998*, pages 111–122, 1998.

[31] Jianqiu Xu and Ralf Hartmut Güting. A generic data model for moving objects. *GeoInformatica*, 17(1):125–172, 2013.

9. Anexo

La implementación fue realizada en Secondo como se menciona anteriormente, por lo cual todo el código que involucra la implementación es escrito en C++.

El código puede ser encontrado en : `https://github.com/meraioth/IDMMO`.