

NGAT : Plataforma para la obtención de Redes de Transporte para el trabajo con Problemas de Diseño de Rutas

NGAT: Framework for obtaining Transport Networks for work with Routes Design problems

Rainer Llera Menocal¹, Alberto Fernández Oliva¹, Joanna Campbell Amos¹

Resumen La sociedad actual depende en gran medida de la optimización de sus procesos. Entre estos se encuentra, el uso de las redes de carreteras, líneas eléctricas, caminos, etc. La correcta representación de dichas redes haciendo uso de la Teoría de Grafos es, por ende, un aspecto de vital importancia en diversos contextos del mundo actual. El presente trabajo brinda un marco de trabajo (*framework*) que constituye una solución computacional para representar redes de transporte de forma novedosa, eficiente y flexible en correspondencia con el desarrollo actual de las tecnologías de la información y las comunicaciones. El resultado que se presenta, sirve, por tanto, como enlace entre los sistemas de información geográfica, proveedores de la información sobre las redes antes mencionadas y los sistemas computacionales diseñados para la solución de problemas de optimización relacionados con las redes, como, por ejemplo, los problemas de rutas.

Abstract Today's society depends to a large extent on the optimization of its processes. Among these is the use of road networks, power lines, gas lines, etc. The correct representation of such networks using Graph Theory is, therefore, an aspect of vital importance in diverse contexts of the world. The present work provides a framework that constitutes a computational solution to represent transport networks in a novel, efficient and flexible way in correspondence with the current development of information and communication technologies. The result, therefore, serves as a link between the geographic information systems, information providers on the aforementioned networks and the computer systems designed for the solution of related optimization problems.

Palabras Clave

Redes de Transporte, Problemas de diseño de rutas , SIG, Plataforma

Keywords

Transport Networks, Desing Routing Problems, GIS, Framework

¹ Departamento de Programación e Ingeniería de software, Universidad de la Habana, La Habana, Cuba, afdez@matcom.uh.cu, joanna@matcom.uh.cu

*Autor para Correspondencia, Corresponding Author

Introducción

Muchos problemas reales de la sociedad involucran la búsqueda de caminos a través de calles, carreteras u otro medio de desplazamiento. Así sucede, por ejemplo, en la recogida de residuos sólidos, la distribución de mercancías, inspección de las líneas eléctricas, tuberías de líneas de gas o petróleo, lectura de los contadores eléctricos o de agua, y campañas contra vectores de enfermedades transmisibles[1]. El objetivo fundamental de la solución de muchos de estos problemas es la construcción o el diseño de rutas óptimas, es decir, que su recorrido minimice algún costo o maximice alguna ganancia. Dicha construcción está sujeta, generalmente, a restricciones sobre las operaciones posibles a realizar. Para resolver este tipo de problemas es necesario que se realice la modelación

de las redes de calles, carreteras, líneas eléctricas, etc., de manera confiable. Dichos modelos representan información fiel a la realidad, por lo que su diseño correcto y eficiente es de vital importancia, económica y social, en la actualidad. Como se ha explicado, los problemas de búsqueda o diseño de rutas están vinculados a problemas reales de la sociedad. Para conformar los mismos mediante modelos precisos, se requiere de su interacción con los Sistemas de Información Geográfica (SIG), pues estos brindan la información necesaria referente a las redes en las que se resuelven estos problemas. Existe una gran variedad de problemas que involucran la búsqueda de rutas y el uso de modelos que representan las calles, carreteras, etc. Dichos problemas se pueden clasificar en dos grupos fundamentales atendiendo a su naturaleza: los

Problemas de Enrutamiento de Vehículos (*Vehicle Routing Problems-VRP*) y los Problemas de Enrutamiento por Arcos (*Arc Routing Problems-ARP*). El conocimiento de la estructura que caracteriza a dichos problemas permite un acercamiento a la naturaleza general de los modelos de representación que involucran redes de caminos.

1. Motivación , justificación y formulación del problema

Una primera aproximación a la dimensión del problema de la construcción de grafos para Redes de Transporte, permite darnos cuenta de su importancia y trascendencia pues como ya se ha mencionado, dicho problema tiene muchas aplicaciones en casos reales del mundo actual. Si la solución óptima de problemas vitales para la sociedad depende de una representación eficiente de la Red de Transporte mediante modelos matemáticos basados en la Teoría de Grafos, entonces resulta innegable la necesidad de precisión, correctitud y eficiencia en el proceso de construcción de dichos Grafos. En general, existen diversas plataformas de desarrollo de software que permiten la creación de Grafos a partir de la información geográfica brindada por los SIGs. Por solo citar algunos ejemplos, pueden mencionarse: Rutas [4], *Open Source Routing Machine* [5] e Itinero [6]. Desafortunadamente, algunas de estas plataformas no están disponibles libremente y existen bajo estrictas licencias comerciales de software propietario, que pueden pertenecer o no a los proveedores de los SIGs, mientras que otras no cumplen con todas las expectativas para la construcción de Grafos que representen Redes de Transporte independientemente de su uso final. Por ejemplo Rutas tiene las siguientes carencias:

- En la documentación disponible no se hace referencia a la manera en la que el software construye el Grafo para representar la Red de Transporte.
- No se especifica si el *software* está desacoplado con respecto al origen del cual procede la información geográfica que procesa (solo se refieren al SIG como una entidad en general).
- No se brinda información en lo relativo a la persistencia de las estructuras de datos que se obtienen (*Grafos*).
- Esta propuesta, independientemente de las buenas prácticas que pudieron ser empleadas o no en su desarrollo, tiene un alcance bastante limitado en cuanto a su utilización. Es decir, solo puede ser utilizada como *software* de escritorio. Por tanto, no puede ser reutilizado, por ejemplo, como librerías ni por terceros.

En el caso de Itinero encontramos las siguientes carencias:

- Como librería, carece de una documentación explícita. La misma se limita solo a comentarios que aparecen en el código fuente del proyecto. Lo anterior obliga a los usuarios del *software* a estudiar dicho código para usos más complejos que los mostrados en los ejemplos. .

- Hasta este momento, su implementación está orientada, específicamente, a problemas simples de rutas y no contempla problemas de optimización más complejos asociados a las Redes de Transporte como, por ejemplo, los ARP. No obstante, actualmente existe una iniciativa, incipiente aun, para abarcar este tipo de problemas
- Itinero está obligada a utilizarse solo como librería y no como un servicio externo para su explotación por terceros ya que carece de un sistema de APIs para tales fines.
- En cuanto a la persistencia y exportación de los datos, solo se tienen en cuenta los que sus autores denominan RouterDB (o base de datos de rutas). Estos son archivos en un formato muy particular que, aunque contiene toda la información referente a la Red de Transporte (el Grafo), no pueden ser utilizados directamente por algún cliente del *software* (otra librería) directamente. Itinero no posee ningún mecanismo interno para exportar el Grafo que representa la Red de Transporte a algún formato estándar.

Las principales desventajas de NetworkX que se le pueden señalar son:

- Aunque es un proyecto bastante flexible (hereda las ventajas multiplataforma del lenguaje de programación en el que fue concebido), al no haber sido diseñado específicamente para el trabajo con problemas complejos de optimización (por ejemplo, los VRP), actualmente no es extensible para su uso en la solución de tales tipos de problemas. No obstante, se puede aprovechar del mismo las bondades que ofrece para la representación de Redes de Transporte.
- Si bien puede señalarse que una de sus ventajas es la compatibilidad que posee con el formato de proveedores de información geográfica Shapefile, resulta una desventaja que dicha compatibilidad sea solo con dicho formato. Otros proyectos independientes han sido desarrollados por miembros de la comunidad para tratar de cubrir esta deficiencia. Los mismos, por lo general, hacen uso de las estructuras de datos que brinda NetworkX, y permiten la compatibilidad con otros formatos de información geográfica como OpenStreetMap[13].
- Carece de alguna implementación o extensión para su explotación remota a través de algún sistema de APIs. No se encontraron evidencias de algún proyecto que la utilice en este sentido.

OSRM tiene como principales desventajas:

- Además de las ventajas que presupone el uso del modelo edge-expanded graph su uso trae como desventaja que aumenta el tamaño de la estructura que representa la Red de Transporte [14].
- Esta plataforma utiliza solamente los datos provenientes de OpenStreetMap y su implementación está atada completamente a dicho formato de datos [5]. Esto limita, en cierta medida, su reusabilidad en caso que sea

necesario trabajar con otro proveedor de información geográfica.

- Hasta el momento, no se ha propuesto ninguna extensión de OSRM para su aplicación en la solución de problemas de optimización más complejos como los ARP. Esto limita lo su uso en la solución de problemas de rutas más simples (de punto a punto en general).

Esto presupone la necesidad de desarrollar plataformas de software libre, extensibles y mantenibles, que permita a los investigadores que tengan que resolver problemas de optimización sobre la red de transporte, por ejemplo, un problema de diseño de rutas, no tener que procesar los mapas digitales, convertirlos a Grafos para entonces, posteriormente, poder resolver el problema en particular que les ocupa. Es decir, mediante el uso de las plataformas antes mencionadas, pueden abstraerse de dichas tareas y podrán explotar directamente el Grafo que se deriva de la información procesada automáticamente. Teniendo en cuenta también, que la información que se maneja en los SIG y por tanto en la solución de problemas de diseño de rutas, equivalen a grandes volúmenes de datos, cuyo procesamiento es computacionalmente costoso, resulta trascendente garantizar la eficiencia en cuanto a: persistencia y formas de representación. A partir del estudio del estado del arte realizado en relación al problema de la construcción de Grafos para Redes de Transporte, se observa que, actualmente, existen diversos contextos de aplicación, asociados a problemas de optimización de procesos, donde los investigadores requieren de una Red de Transporte para representar en ella la información asociada a dicho problema y la cual fue suministrada por cualquiera de los proveedores de información geográfica existentes. Sin embargo, el formato en que viene esta información tiene que ser transformado a un formato que pueda ser procesado por los algoritmos de optimización que serán utilizados para dar solución al problema. Por tanto, es necesario concebir plataformas computacionales que de forma automatizada hagan tal conversión. Teniendo en cuenta el contexto de la investigación, se puede definir el problema identificado de la siguiente forma: aunque en la actualidad existen herramientas orientadas a este propósito, no se encontró en la comunidad *Open Source* una plataforma abierta lo suficientemente extensible, reutilizable y abstracta que simplemente procese la información geográfica existente y obtenga a partir de ella un Grafo (como representación de la Red de Transporte) sin importar el proveedor que suministró la información ni el contexto donde será utilizado.

1.1 Objetivos

El objetivo general de este trabajo es desarrollar un componente, o biblioteca de código, cuyo propósito sea construir Grafos para Redes de Transporte, a partir de mapas digitales brindados (en su mayoría) por los SIG's y persistir, o exportar, dichos grafos de manera flexible para su uso por componentes de terceros. Para garantizar el cumplimiento del objetivo general planteado, se propone alcanzar, paulatinamente, un conjunto de objetivos específicos. Son los siguientes:

1. Garantizar abstracción en cuanto al formato en el cual viene la información geográfica suministrada.
2. Garantizar abstracción de la capa de datos en cuanto a la tecnología y la representación de los mismos.
3. Implementar una arquitectura de *software* para poder extender la funcionalidad de la plataforma en lo que respecta a resolver problemas de diverso tipo asociados a la Teoría de Grafos u otro cualquier aspecto inherente a la plataforma que pueda ser extensible.
4. A partir del objetivo anterior, permitir flexibilidad en cuanto al tipo de información que se importa del mapa digital para construir el Grafo. Esto facilita que los clientes de la plataforma puedan decidir, según sus necesidades, qué datos procesar en cada momento.
5. Implementar un procesador concreto para uno de los posibles formatos de mapas de datos (Open Street Map ha sido el seleccionado).
6. Implementar un algoritmo concreto que muestre la extensibilidad expresada en 3. (el algoritmo de caminos de costo mínimo de Dijkstra ha sido el seleccionado).
7. Crear un sistema API con tecnología *RESTful* que permita el uso de la plataforma para la creación y explotación de Grafos de manera remota.

2. Problemas de Rutas

Los problemas que involucran la búsqueda o selección de caminos para satisfacer una determinada demanda o servicio, como los VRP y ARP, poseen elementos comunes que son independientes de la naturaleza de los mismos Figura 1. Entre ellos, los más representativos son los siguientes:

1. Vehículo: En este tipo de problemas la definición de un vehículo debe verse desde una perspectiva lo más general posible, o sea, en dicho contexto un vehículo podría ser, una señal eléctrica que transporta información, un ser humano que transporta una determinada carga, un avión, un barco, un auto, una bicicleta, un animal, etc. Los vehículos se definen a partir de un conjunto de atributos, tales como, su capacidad de carga (peso y/o volumen), costes asociados a la utilización del mismo, su autonomía, etc. Resulta evidente, a partir de lo expresado, que en los mismos se puede transportar diferentes tipos de cargas.
2. Flota de vehículos: No es más que un conjunto de vehículos de igual o diferente tipo. Cuando todos los vehículos de la flota comparten las mismas características, entonces, se dice que la flota es homogénea. En caso contrario, se dice que la flota es heterogénea. El número de vehículos asociado a una flota puede ser conocido, o puede ser un valor a decidir.
3. Red de transporte: es un conjunto de entidades específicas, conectadas entre sí, que permite que por ella circulen los vehículos que transportan ciertas cargas, materiales o inmateriales, entre estas entidades, según reglas bien definidas. Estas entidades podrían ser, por

ejemplo, almacenes, conectados por carreteras o, switches conectados por cables eléctricos, etc. El modelo matemático a partir del cual se describe una red de transporte es un Grafo.

4. Depósitos centrales o Almacenes: son otros de los posibles componentes de la red (nodos del grafo también) los cuales se definen a partir de su localización, además de otras características que los diferencian entre sí. Por ejemplo, capacidad máxima de almacenamiento, horario de disponibilidad, etc.
5. Ruta: es una secuencia de nodos pertenecientes a la red de transporte, que puede ser recorrida en el orden dado, es decir, que existe un enlace directo entre los pares de nodos consecutivos. Puede entenderse como un camino establecido dentro de la red de transporte. En el contexto de los problemas de búsqueda de rutas, a partir de estas se optimiza el costo necesario, o la ganancia obtenida de la satisfacción de una determinada demanda. Por ejemplo, minimizar el gasto de la cantidad total de combustible usado por una flota de vehículos que deben brindar un determinado servicio, o minimizar la cantidad de vehículos de la flota para dar dicho servicio.

Generalmente, la solución de los problemas de rutas, tienen como entrada, una representación de la red de transporte (grafo) y dan como salida, una o varias rutas a seguir. En la literatura consultada, no existe una definición formal y abarcadora para el término Problema de Ruta. A los efectos de este trabajo, hemos asumido la siguiente definición: Problema de Rutas (RP): es aquel que involucra el diseño o selección de rutas dentro de una red de transporte, con el objetivo de optimizar algún proceso. Como se explicó anteriormente, los RP se pueden clasificar en dos grupos (VRP y ARP) que, aunque comparten los elementos anteriormente descritos como parte de su definición, difieren en cuanto a los contextos donde son usados ya que el objeto de estudio es diferente, en un caso son los nodos y en el otro los enlaces.

3. Sistemas de Información Geográfica (SIG)

Un Sistema de Información Geográfica (SIG) es un conjunto de herramientas que integra y relaciona diversos componentes (usuarios, hardware, software y procesos) que permiten la organización, el almacenamiento, la manipulación, el análisis y la modelación de grandes cantidades de datos procedentes del mundo real. Dichos datos están vinculados a una referencia espacial, esto facilita la incorporación de aspectos sociales-culturales, económicos y ambientales a los mismos con el objetivo de hacer más eficiente la toma de decisiones[2]. El SIG funciona como una base de datos con información geográfica en la que los objetos (provenientes de mapas digitales) que se encuentran almacenados en ella se asocian entre sí a partir de un identificador común (establece una geo referencia). De esta forma, dado cualquier objeto almacenado en la base de datos, podemos conocer su loca-

lización geográfica e, inversamente, dada una localización geográfica podemos conocer todos los objetos y atributos geográficos asociados a ella[3]. Los SIGs almacenan información procedente del mundo real como pueden ser, por ejemplo, la referida a carreteras, ríos, lagos, montañas, etc. Existen dos formas principales en las que se representan y se almacenan los datos antes mencionados [7]:

- raster o de retícula.
- vectorial.

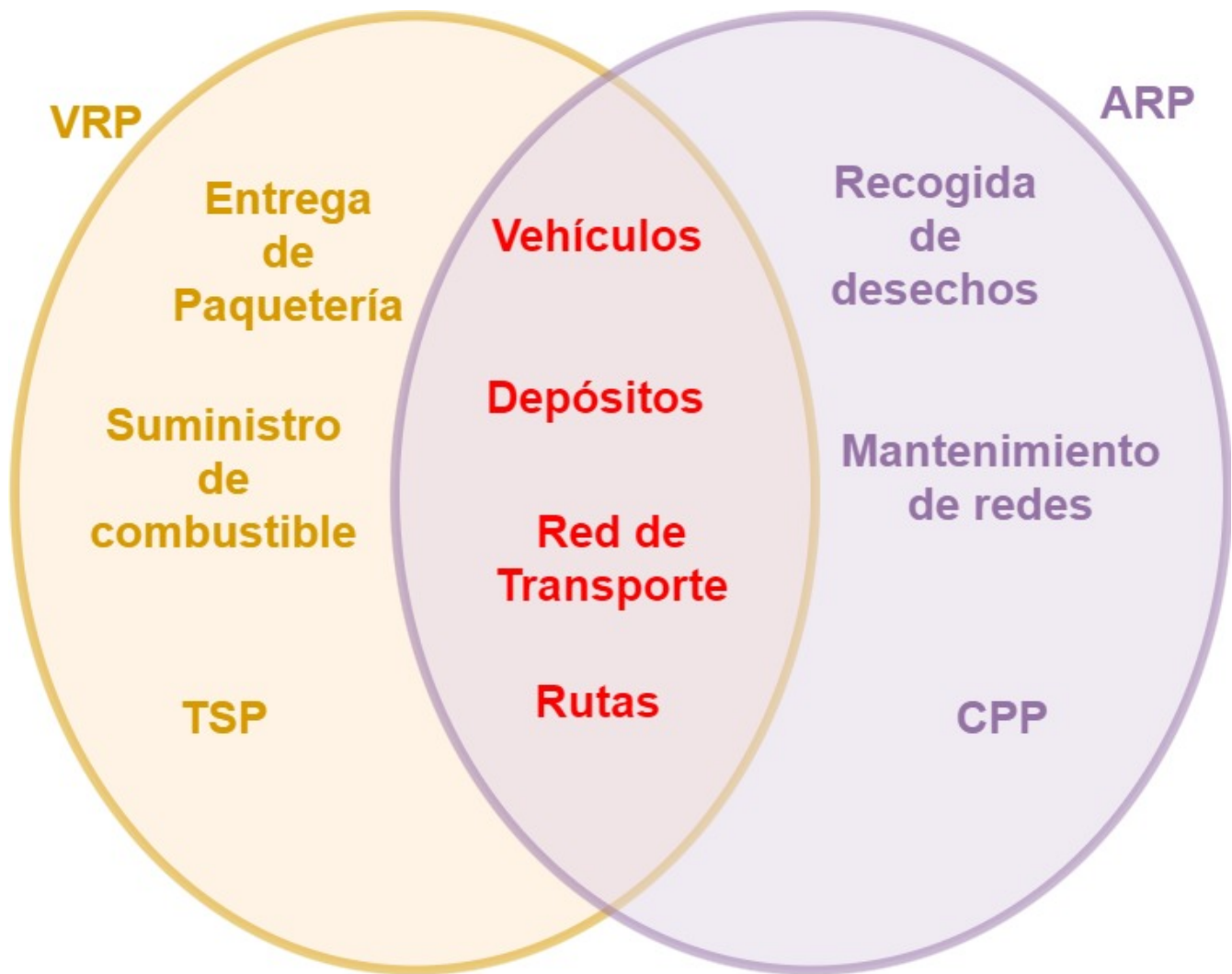
El modelo vectorial modela los objetos geográficos de interés utilizando tres elementos geométricos fundamentales: Puntos, Polilíneas, Polígonos . Ver Figura 2

4. Integración de la información geográfica con la Teoría de Grafos

Teniendo en cuenta lo expuesto anteriormente, resulta evidente que para la construcción de grafos que representen Redes de Transporte es necesario interpretar la información proporcionada por los SIGs en el modelo vectorial. Dado que estas redes se construyen, por lo general, sobre la red de carreteras y caminos de las ciudades y localidades, para su modelación mediante un grafo, el primer paso que habría que realizar es filtrar las geometrías de interés, es decir, determinar las polilíneas que cumplan ciertos criterios a partir de ciertos valores de sus atributos (por ejemplo, que estén marcadas como calles, para evitar incluir los ríos). En la práctica, la secuencia de puntos que definen una polilínea, puede representar diferentes entidades entre las cuales, se incluyen: intersecciones con otras polilíneas (calles, arroyos, líneas de ferrocarril, etc.), objetos de interés como, por ejemplo, una señal de tránsito, un transformador en las vías eléctricas, o un cambio considerable en la dirección de la vía. Teniendo en cuenta que una polilínea es una secuencia de puntos, se debe describir como se representa la misma en el Grafo:

- Cada punto de la polilínea es un vértice del Grafo.
- Entre cada par de puntos (vértices) consecutivos de la polilínea se establece una arista (cuando no hay ningún atributo asociado a la polilínea que indique dirección de circulación) o un arco, en caso contrario. Por tanto, el objeto resultante, teniendo en cuenta lo antes descrito es, generalmente, un grafo mixto. Teniendo en cuenta las complejidades que presenta el mundo real, se puede decir que el modelo que se construye es un multígrafo mixto ponderado, pues en el mismo pueden coexistir: arcos, aristas y lazos. Además, los arcos o aristas, tienen un costo asociado, el cual, en su forma más simple, puede ser la distancia entre los puntos (vértices) que las definen.

Para diversas situaciones del mundo real, modelar la Red de Transporte, bajo dicha concepción, puede dar como resultado un grafo muy complejo que, por lo general, resulta difícil de procesar por los algoritmos de optimización existentes. Por



Problemas de Rutas

Figura 1. Elementos comunes de los problemas de rutas

tanto, resulta evidente que es necesario alcanzar simplificaciones ante este tipo de grafos. Los tipos de simplificaciones que se pueden asumir pueden ser tan diversas como el tipo de situaciones que pueden presentarse, por tanto, dichas simplificaciones no pueden ser generalizadas. Expliquemos lo anterior mediante un ejemplo: En el caso específico de las redes de calles, la modelación se debe hacer lo más genérica posible, ya que pueden existir restricciones sobre dichas redes, como, por ejemplo, las regulaciones del tránsito, que limiten el acceso a las mismas en dependencia del vehículo que circule por ella. Resulta evidente que una calle de un solo sentido de dirección, limita la circulación por ella a los vehículos automotrices, pero no a los peatones que la pueden circular por ella en ambos sentidos haciendo uso de sus aceras.

5. NGAT: Framework para el análisis de grafos de redes

Network Graph Analysis Tools (NGAT) es la propuesta de este trabajo para dar solución al problema planteado en el apartado introductorio. El objetivo esencial del marco de trabajo que se propone es permitir la integración de los datos brindados por los SIGs con las soluciones a los problemas que se generan en las Redes de Transporte. La plataforma de trabajo, o framework, ha sido desarrollada en el lenguaje de programación C# y se utilizó como *Integrated Development Environment (IDE)* para su desarrollo, *Microsoft Visual Studio 2017 Community Edition*. La implementación de NGAT está centrada, principalmente, en dos modalidades de uso:

- Como plataforma (librerías) para la interpretación de


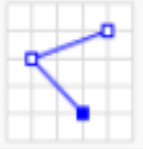

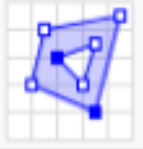
Type	Ejemplos	
Punto		<code>POINT (30 10)</code>
Polilínea		<code>LINESTRING (30 10, 10 30, 40 40)</code>
Polígono		<code>POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))</code>
		<code>POLYGON ((35 10, 45 45, 15 40, 10 20, 35 10), (20 30, 35 35, 30 20, 20 30))</code>

Figura 2. Objetos geométricos del modelo vectorial

datos geográficos y su persistencia como grafos de redes

- Como servicio web para la ejecución de algoritmos sobre dichos grafos..

5.1 Arquitectura de NGAT

NGAT está desarrollada siguiendo una arquitectura multicapas [8]. La decisión de implementar este proyecto siguiendo dicho patrón se debe, en primer lugar, a la facilidad que este modelo supone para el desarrollo de herramientas de software. Esta facilidad, en esencia, se traduce en lo siguiente: independientemente de la aplicación que se esté diseñando, resulta intuitiva la descomposición del diseño mediante el agrupamiento lógico de componentes de software [9]. Si el modelo teórico inherente a la arquitectura multicapas se implementa correctamente, entonces, la misma sigue los principios SOLID[11] de la programación orientada a objetos y a partir de esto, se garantizan los principios de extensibilidad y mantenibilidad que son objetivos esenciales en el presente trabajo. Teniendo en cuenta los requerimientos de NGAT, en el proceso de diseño de las capas lógicas que lo conforman, se determinó que era necesario tener, al menos, tres capas:

- Capa de negocio .
- Capa de servicios.
- Capa de acceso a datos.

Además, se implementó una capa de infraestructura transversal para contener las funcionalidades comunes y necesarias para el resto de las capas. En la Figura 3 se muestra un esquema que describe la arquitectura multicapas de NGAT. Para la

implementación de esta plataforma, se siguió el principio de diseño guiado por el dominio (*domain-driven design, DDD*) [12]. Este enfoque plantea, básicamente, que el diseño del software que se crea, debe girar en torno al dominio del problema que se pretende resolver. En este contexto, el término dominio se interpreta como todos los requisitos que debe cumplir el software para satisfacer las necesidades del usuario. Esta filosofía de diseño permite, por tanto, la separación de la lógica del negocio de otros aspectos relacionados con el software en sí. Para el desarrollo de las funcionalidades iniciales que se brindan como parte de NGAT, se siguió un diseño guiado por pruebas (*test-driven design, TDD*)[10]. Este enfoque garantiza la correctitud de dichas funcionalidades y optimiza el proceso de desarrollo. Bajo esta concepción, primeramente, se diseñan las pruebas que deben pasar las unidades de código y, posteriormente, se implementan las mismas, atendiendo a los resultados alcanzados en dichas pruebas. Entre las ventajas que brinda el diseño de NGAT se resaltan las siguientes:

- Se alcanza un nivel de independencia en la plataforma que permite la sustitución de alguna, o algunas, de las capas dependientes de la capa de negocio (core de la aplicación) por capas nuevas que se diseñen sin que se vea afectado el funcionamiento del resto del sistema. O sea, el añadir o sustituir unas capas por otras, no afecta la funcionalidad de la plataforma. La Figura 3 muestra la dependencia entre las capas de la plataforma. Como se aprecia en dicha figura, la capa de aplicación, así como la de acceso a datos dependen, exclusivamente, de la capa de negocios. Esta es, precisamente, la metodología

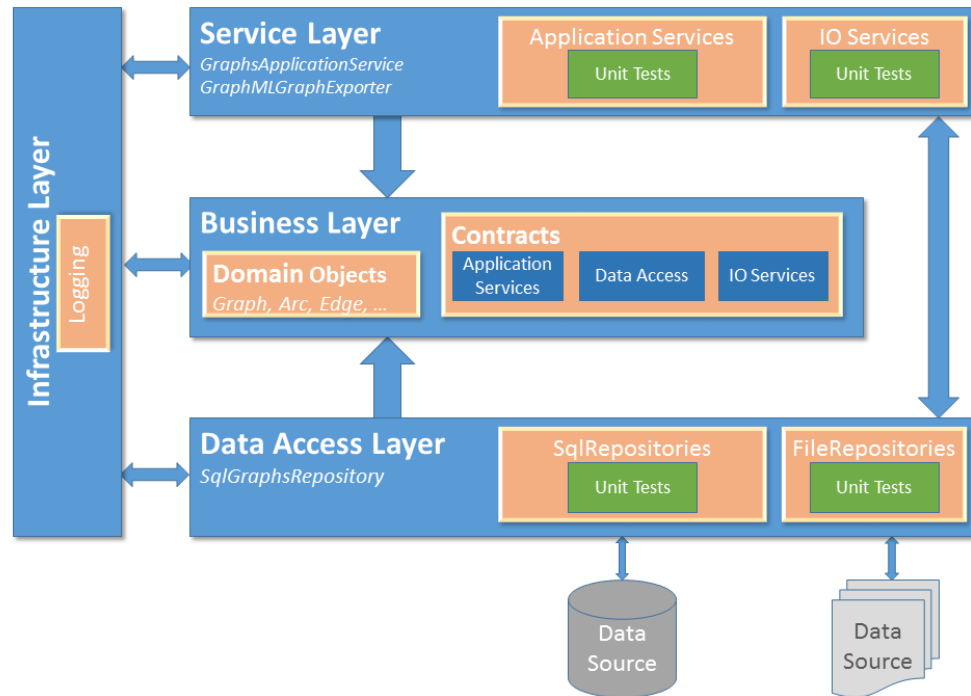


Figura 3. Arquitectura de NGAT

que define el DDD.

- La extensibilidad está implícita en este tipo de diseño: si se requiere extender la funcionalidad de la plataforma, a partir de los contratos definidos, se puede implementar la misma en la capa correspondiente (acceso a datos o servicios). Si se incluyen nuevos objetos al dominio, bastaría con declarar los nuevos contratos referentes a la lógica de negocio que implican dichos objetos, e implementar dichos contratos en las capas correspondientes

5.2 Flujo de trabajo con el framework

Para la utilización de NGAT como framework (librerías), el flujo de trabajo de los clientes del software puede ser resumido en dos variantes principales.

- Trabajo sin persistencia de los datos: Esta variante supone que no se utilizarán las facilidades de persistencia de los datos que implementa la capa de acceso a los mismos. Puede resultar útil cuando se trabaja con problemas de pequeña dimensión y no se requiere del almacenamiento de los objetos de tipo grafo. Para trabajar con esta metodología, un cliente de la plataforma debe utilizar la capa de servicios, la componente de entrada/salida. De esta forma, se tendrá acceso a las funcionalidades referidas a la construcción de los grafos e importación y exportación de los mismos a diferentes formatos.
- Trabajo con persistencia de los datos: Al trabajar con la persistencia de los datos, los clientes de la plataforma

pueden hacer uso de las facilidades expuestas en el punto anterior

6. Resultados y Experimentación

Para la validación de NGAT se realizaron varias implementaciones que demuestran su utilidad en el proceso de construcción de grafos que representan redes de transporte. Los primeros medidores utilizados para dicha validación, fueron las pruebas unitarias realizadas a las funcionalidades implementadas. En la Figura 4 se muestra una vista parcial del grafo de red construido en las pruebas realizadas al objeto PBFGraBuilder, encargado de procesar los archivos en formato PBF de OpenStreetMap. Los nodos de dicho grafo se ven como marcadores y las líneas grises representan los enlaces entre dichos nodos. La visualización de los grafos construidos mediante las pruebas unitarias, y la experimentación con dichos grafos, dieron lugar a algunas optimizaciones en cuanto a la representación de los mismos. Como ejemplo de lo anterior, se puede mencionar la eliminación de puntos del grafo cuyo único propósito es marcar cambios de dirección. De esta manera, se simplifica la representación inicial y no se pierde la información relacionada con dichos puntos. La geometría de los mismos se almacena como atributo de los enlaces.

Además de las pruebas unitarias, se implementaron tres aplicaciones clientes de la plataforma:

- NGATTools: es una aplicación que utiliza la API de la plataforma y presenta sus funcionalidades mediante

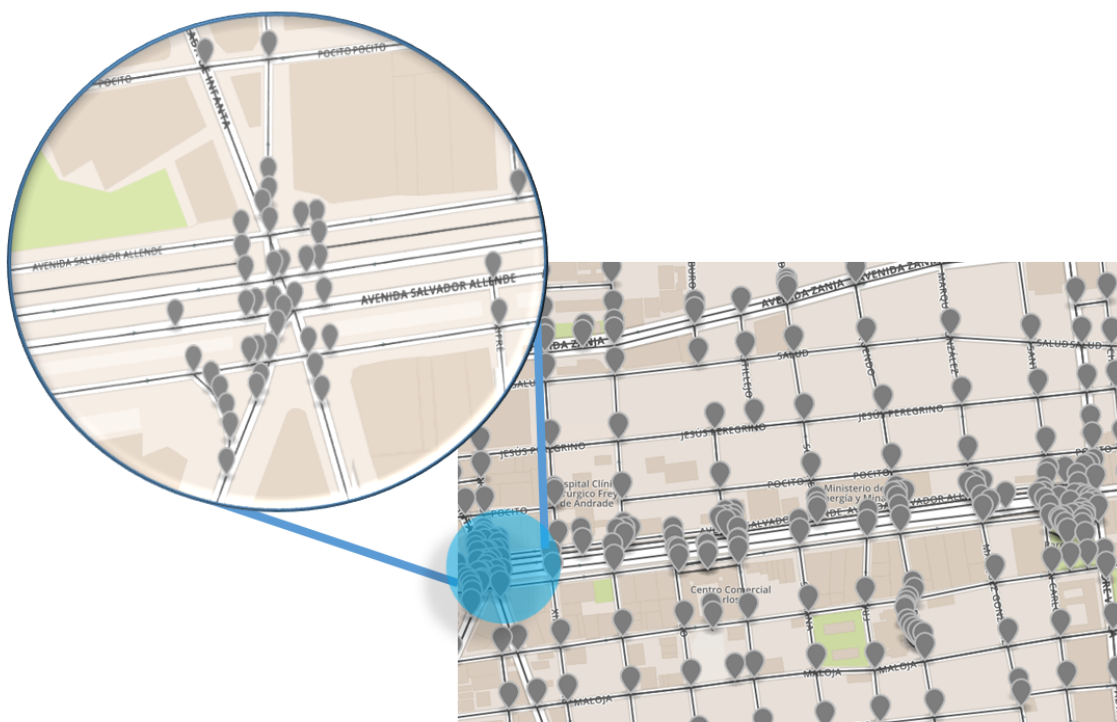


Figura 4. Grafo de red parcial de La Habana. A la izquierda, intersección de la Avenida Salvador Allende con la Calzada de Infanta.

comandos de consola. A través de esta aplicación, se pueden construir los grafos (o subgrafos) desde los proveedores implementados (OpenStreetMap), exportar los mismos a los diferentes formatos soportados por NGAT (GeoJSON y GRF) y ejecutar algoritmos sobre los grafos construidos. Su implementación permite la utilización del framework por sistemas de terceros que requieran solamente de la representación de las redes de transporte en algún formato específico.

- NGAT.WebAPI: puede verse como una aplicación cliente de NGAT o como una capa de servicios más de esta plataforma, ya que es un servicio web que, al igual que NGATTools, permite usar las funcionalidades de la plataforma mediante pedidos HTTP. Tiene como principal ventaja, el aislamiento de la plataforma de sus clientes finales. A través del protocolo HTTP, diferentes usuarios, independientemente de su origen (sistema operativo, dispositivo, etc.), pueden hacer uso de las facilidades que brinda NGAT. Además, abre las puertas para posibles capas de presentación de la plataforma, aisladas de la lógica de las capas restantes, que podrían comunicarse e interactuar con el resto de la plataforma mediante el uso de este servicio web.
- TravelAgencyMVC: es una aplicación web desarrollada en la Universidad de La Habana. Se presenta como cliente de NGAT.WebAPI. Utiliza sus funcionalidades a partir de la interacción con los usuarios finales de la referida aplicación. A través de ella, se puede selec-

cionar en un mapa de Cuba alguna región, construir el subgrafo referente a la misma y ejecutar algún algoritmo, utilizando NGAT.WebAPI. Además, se permite visualizar dicho subgrafo (o la respuesta del algoritmo ejecutado) utilizando OpenLayers (OpenLayers.Org, 2017). Esta aplicación muestra la posible utilización de NGAT como framework para clientes completamente externos de la plataforma, o sea, clientes para los cuales no es posible la utilización de la misma como una biblioteca de código (DLLs). La Figura 5 5 muestra una captura de esta aplicación en el proceso de mostrar una ruta calculada mediante el algoritmo de Dijkstra.

7. Conclusiones

En este trabajo se presentó una propuesta de desarrollo de una plataforma de software cuyo propósito es construir grafos para la representación de Redes de Transporte, a partir de los datos brindados en mapas digitales por los diversos proveedores de información geográfica. De forma más detallada podemos decir que los principales aportes de la presente investigación, son los siguientes:

Diseño e implementación de un framework con las siguientes funcionalidades:

- Un importador de grafos para el formato PBF de OpenStreetMap.
- Un exportador de grafos para el formato GeoJSON.

- Un importador y exportador, específicos, para el formato GRF.
- Un servicio genérico para aplicar filtros a los elementos de fuentes de datos geográficas.
- Un conjunto de APIs para interactuar con los elementos anteriores, usando una aplicación de consola como cliente.
- Un servicio web cliente del framework (*WebAPI*) que posibilita utilizar el mismo de forma remota.

Además, se realizaron las siguientes implementaciones computacionales:

- Un algoritmo de ruteo (Dijkstra) que demostró las potencialidades del framework.
- Una aplicación cliente, con mapa, que permite visualizar los resultados alcanzados por el algoritmo anterior.
- Operaciones geométricas para la intersección de polígonos con los grafos generados.

Se puede concluir, por lo expuesto anteriormente, que se cumplió el objetivo primario de este trabajo, así como los objetivos secundarios o específicos, mencionados en la sección introductoria.

7.1 Líneas futuras de investigación

Los elementos que permiten dar continuidad inmediata a la investigación y en función de los resultados ya alcanzados, son los siguientes:

- Implementar procesadores para datos que provengan de otros proveedores de información geográfica, tales como, *Shapefile*, *OpenStreetMap XML*, *SQL Server Features*, etc.
- Incorporar algún sistema de *plugins* al servicio web de forma tal que permita extender, dinámicamente, la funcionalidad de la plataforma.
- Optimizar la representación interna de los grafos mediante algoritmos que permitan ordenar el espacio bidimensional como, por ejemplo, la ordenación por la curva de Hilbert, lo cual brindaría una obtención más eficiente de los objetos geométricos que componen el grafo.

Referencias

- [1] Amos, M. J. Herramienta computacional para la resolución del problema del cartero chino para grafos dirigidos, no dirigidos y mixtos. Tesis de Maestría, pp 60, (2013)
- [2] Star, J., Estes, J. Geographical Information Systems: An Introduction. (1990)
- [3] Bolstad, P. GIS Fundamentals: A first text on Geographic Information Systems, Second Edition. White Bear Lake, MN: Eider Press, pp 543, (2005)
- [4] Villalobos, A. R. Integración de un SIG con modelos de cálculo y optimización de rutas de vehículos CVRP y software de gestión de flotas. Madrid. (35), pp 8 (2007)
- [5] OSRM - Backend. OSRM Backend - GitHub. Retrieved from <https://github.com/Project-OSRM/osrm-backend> (2017)
- [6] tinero. Itinero. Retrieved from <http://www.itinero.tech> (2017)
- [7] Tomlin, C. D. Geographic information systems and cartographic modeling. Prentice Hall. (1990)
- [8] Richards, M. Software Architecture Patterns. O'Reilly. (2010)
- [9] Microsoft MSDN . Microsoft Application Architecture Guide. Microsoft Press. (2009)
- [10] Fowler, M. Unit Test. Retrieved from <https://martinfowler.com/bliki/UnitTest.html>. (2017)
- [11] Metz, S. SOLID Object-Oriented Design. (2009, Enero)
- [12] Evans, E. Domain-Driven Design - Tackling Complexity in the Heart of Software. Addison Wesley. (2004)
- [13] Mapkin, OsmGraph. Retrieved from <https://github.com/Mapkin/osmgraph>, (2017, 3).
- [14] Luxen, D, Smart Directions Powered by OSRM's Enhanced Graph Model. Retrieved from <https://www.mapbox.com/blog/smart-directions-with-osrm-graph-model/>, (2017, 4)