

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

**REDISEÑO DEL SISTEMA C4R (CITYMOB FOR ROADMAPS),
CON LA IMPLEMENTACIÓN DE NUEVAS OPCIONES EN LAS
HERRAMIENTAS DEL MOTOR DE SIMULACIÓN DE TRÁFICO
VEHICULAR SUMO (0.30.0), PARA LA OBTENCIÓN DE
SIMULACIONES MÁS REALISTAS**

**TESIS DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE MAGÍSTER EN
CONECTIVIDAD Y REDES DE TELECOMUNICACIONES**

SANTIAGO JAVIER OÑATE CHAVEZ

sonate@redesps.com

DIRECTOR: DR. LUIS FELIPE URQUIZA AGUIAR

luis.urquiza@epn.edu.ec

CODIRECTOR: ING. XAVIER ALEXANDER CALDERON HINOJOZA, MSc.

xavier.calderon@epn.edu.ec

Quito, Agosto 2018

AVAL

Certificamos que el presente trabajo fue desarrollado por Santiago Javier Oñate Chávez, bajo nuestra supervisión.

DR. LUIS FELIPE URQUIZA AGUIAR
DIRECTOR DEL TRABAJO DE TITULACIÓN

ING. XAVIER CALDERON HINOJOZA, MSc.
CODIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo Santiago Javier Oñate Chávez, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

SANTIAGO JAVIER OÑATE CHAVEZ

DEDICATORIA

Esta tesis va dedicada a mi esposa, a mi futuro hijo y mis padres.

A Ellos.

AGRADECIMIENTO

Quiero agradecer principalmente a Dios por darme la vida, a mi esposa por ser un apoyo para la culminación de este proyecto, a mis padres por darme la vida, guiarme y enseñarme lo importante, a mis familiares y amigos por estar siempre para apoyarme en cualquier momento, a Luis Felipe y Xavier por su ayuda y dirección en la realización de este proyecto, no debo dejar de mencionar a mi peludo amigo, que me acompaña con su alegría.

Gracias.

ÍNDICE DE CONTENIDO

AVAL	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	X
ABSTRACT	XI
1. INTRODUCCIÓN.....	1
1.1 Pregunta de Investigación.....	2
1.2 Objetivo General	2
1.3 Objetivos Específicos	2
1.4 Alcance	3
1.5 Estructura del proyecto	3
1.6 Marco Teórico	4
1.6.1 Redes Adhoc y Redes VANETs.....	4
1.6.2 Ventajas y desventajas de las redes Ad-hoc.....	5
1.6.3 Dominio en una VANET.....	7
1.6.4 Características de las redes Vehiculare.....	8
1.6.5 Vehículos Inteligentes.....	9
1.6.6 Aplicaciones y modelos de movilidad	10
1.6.7 JmapViewer.....	10
1.6.8 Open Street Maps	11
1.6.9 NETCONVERT.....	12
1.6.10 POLYCONVERT	13
1.6.11 SUMO (Simulation of Urban Mobility)	14
1.6.12 Definición de Modelos de Movilidad de Vehículos (Car-Following Models).18	
1.6.13 Estudio y análisis de los modelos de Movilidad para su uso.....	22
2. METODOLOGIA	26
2.1 Identificación del problema, oportunidades y objetivos	26
2.2 Objetivos y oportunidades para el desarrollo	26
2.3 Análisis y necesidades de C4R.....	27
2.4 Diagramación UML del Software.....	31

2.4.1	Fase de Inicio	31
2.4.2	Requerimientos Específicos	31
2.4.3	Confiabilidad.....	32
2.4.4	Desempeño	32
2.4.5	Consultas, Informes y entregables.....	32
2.5	Modelos de Casos de Uso del software	33
2.6	Diagrama de casos de uso.....	34
2.7	Diagrama de Componentes	37
2.8	Diagrama de navegación	39
2.9	Diagrama de Clases.....	40
2.10	Codificación y programación del módulo de simulación	41
2.10.1	Generación de Simulaciones con parámetros aleatorios	41
2.10.2	Generación de Simulaciones con parámetros personalizados	43
2.10.3	Manejo de archivos OSM descargados con JMapView	43
2.11	Algoritmos y clases para la programación	45
2.11.1	Algoritmos de búsqueda en archivos	46
2.11.2	Algoritmos de creación de archivos	48
2.11.3	Algoritmos de ejecución de archivos.....	51
2.12	Requisitos de Software y Hardware	52
2.12.1	Requisitos de Software.....	52
2.12.2	Requisitos de Hardware	52
2.12.3	Consideraciones para Memoria y Procesador	53
	Consideraciones de espacio en disco duro.....	54
2.13	Manual de Usuario del programa C4R Mejorado.....	54
3.	RESULTADOS Y DISCUSIÓN	64
3.1.	Resultados.....	64
3.2.	Discusión.....	75
4.	CONCLUSIONES	77
5.	REFERENCIAS BIBLIOGRÁFICAS	80
6.	ANEXOS.....	82
	ANEXO I. CODIGO FUENTE DEL PROGRAMA	83
	ANEXO II. ARCHIVO DE RUTAS GENERADO POR SOFTWARE MEJORADO	125
	ANEXO III. ARCHIVO DE RUTAS GENERADO POR SOFTWARE C4R EN VERSION ORIGINAL	132
	ORDEN DE EMPASTADO	142

ÍNDICE DE FIGURAS

Figura 1.1. Dominios y componentes de una VANET	8
Figura 1.2 Sección de archivo OSM.xml descargado por JMapView	12
Figura 2.1 Diagrama de casos de Uso General	34
Figura 2.2 Diagrama de componentes que muestra la dependencia en tiempo de ejecución.	37
Figura 2.3 Diagrama de componentes que demuestra la dependencia en tiempo de compilación	38
Figura 2.4 Diagrama de navegación	40
Figura 2.5 Porción de código para ejecución de randomtrips.py	42
Figura 2.6 Ejecución real de comando de Python	42
Figura 2.7 Porción de código del archivo OSM	44
Figura 2.8 Ejemplo de relación de nodos en archivo OSM	44
Figura 2.9 Porción de código de algoritmo para leer archivos XML de OSM	47
Figura 2.10 Porción de código para escribir nombre de vía y discriminar tipo de vía	48
Figura 2.11 Porción de código de algoritmo de selección de vía continua	50
Figura 2.12 Porción de código que genera el archivo de rutas personalizado	51
Figura 2.13 Ejecución de NETCONVERT en el <i>Command Prompt</i> de Windows	52
Figura 2.14 Menu principal para realizar una simulación	55
Figura 2.15 Grabar un nuevo proyecto dentro de una carpeta específica	55
Figura 2.16 Habilitación de la opción Seleccionar Mapa	56
Figura 2.17 Selección de mapa para realizar la simulación	56
Figura 2.18 Descarga del archivo OSM	57
Figura 2.19 Opciones de POLYCONVERT para la generación del archivo typemap	57
Figura 2.20 Mensaje de creación correcta de archivo typemap.xml	58
Figura 2.21 Selección de Simulación Aleatoria	58
Figura 2.22 Definición de número de vehículos para una nueva simulación aleatoria	58
Figura 2.23 Selección de Visualización Aleatoria	59
Figura 2.24 Ejecución correcta de parámetros de la simulación	59
Figura 2.25 Ejecución correcta de parámetros de la simulación	59
Figura 2.26 Zoom y ejecución de la simulación de manera aleatoria	60

Figura 2.27 Mensaje de advertencia que la simulación termino correctamente ...	60
Figura 2.28 Selección de simulación con parámetros personalizados	61
Figura 2.29 Cuadro de diálogo para la elaboración de parámetros personalizados	61
Figura 2.30 Creación de un nuevo vehículo	62
Figura 2.31 Creación de una nueva ruta	62
Figura 2.32 Pre visualización de la ruta seleccionada para la generación de rutas	63
Figura 2.33 Generación de archivo de rutas	63
Figura 3.1 Selección de porcion de mapa para realizar simulaciones.....	64
Figura 3.2 Descarga de archivos OSM (Open Street Maps)	65
Figura 3.3 Parametros de TypeMap para PolyConvert	66
Figura 3.4 Generación de archivos para generacion simulación aleatorio	67
Figura 3.5 Simulacion en programa Sumo para estudio	67
Figura 3.6 Parámetros 157005691 vía tiempo 0	71
Figura 3.8 Parámetros vía 157005691 tiempo 100	72
Figura 3.9 Parámetros vía 222299277 tiempo 0	72
Figura 3.10 Parámetros vía 222299277 tiempo 50	73
Figura 3.11 Parámetros vía 222299277 tiempo 100	73
Figura 3.12 Parámetros vía 222299277 tiempo 100	74
Figura 3.13 Simulación Original en SUMO version 0.11	75

ÍNDICE DE TABLAS

Tabla 1.1. Opciones adicionales en definición de vehículos.....	18
Tabla 1.2 Modelos de movilidad.....	24
Tabla 2.1 Posibles casos de uso.....	34
Tabla 2.2 Características Mínimas equipo de computación para Netbeans.....	53
Tabla 2.3 Características mínimas equipo para simulación.....	54
Tabla 3.1 Características y parámetros de vía 157005691.....	67
Tabla 3.2 Características y parámetros de vía 222299275.....	69
Tabla 3.3 Resultados de simulación vía 157005691 en el tiempo.....	70
Tabla 3.4 Resultados de simulación vía 222299275 en el tiempo.....	71

RESUMEN

Las simulaciones en la actualidad son de mucha importancia para poder realizar nuevas investigaciones y crear nuevos conceptos; hacer pruebas en entornos reales a más de ser costosas es casi imposible; por cuanto significaría movilizar una gran cantidad de instrumentos de medición, implementos físicos y personal ocasionando costos muy elevados.

Para que una simulación sea exitosa se necesita que los parámetros considerados en la entrada sean todos o la mayoría de los que intervienen apegados a la realidad, con esto se podría esperar que los resultados sean altamente confiables.

Otro punto importante es que se necesitan que los modelos de movilidad, conceptos y entornos sean los que se manejan en la actualidad (la ciencia siempre está en constante desarrollo y mejoramiento), y en este caso SUMO, y el instituto de Sistemas de Transportación siempre va mejorando y lanzando nuevos conceptos que deben ser usados.

Es por eso, que esta tesis optimizará el software C4R realizado en el año 2011 que utiliza aún versiones de software ya obsoletas para la actualidad y adaptar las versiones actualizadas de SUMO, NetConvert y PolyConvert; dejando la posibilidad de seguir actualizando las nuevas versiones que puedan ser liberadas de estos programas sin perder funcionalidad, ya que se crearon nuevas formas de leer, manipular, y generar archivos para las simulaciones.

Una vez realizados los cambios, se probó el funcionamiento, simulando en mapas reales descargados mediante el servidor de Open Street Maps y el aplicativo JMapView que es un aplicativo que puede acceder y descargar los mapas actualizados de cualquier parte del mundo.

Finalmente, se muestran las conclusiones encontradas y se mencionan recomendaciones para futuros trabajos y proyectos de titulación.

PALABRAS CLAVE: Simulacion, Sumo, Netconvert, Polyconvert, JMapView, Open Street Maps.

ABSTRACT

Simulations are currently of great importance to be able to generate new investigations and create new concepts; testing in real environments, besides being expensive, is almost impossible; because it would mean mobilizing the large number of measuring instruments, physical implements and personnel, causing very high costs.

For a simulation to be successful, it is necessary that the parameters considered in the input be all those that intervene and that these be attached to reality, with this one could expect the results to be almost real.

Another important point is that mobility models, concepts and environments are what are currently being handled (science is always in constant development and improvement), and in this case Sumo, and the Institute of Transportation Systems always it is improving and launching new concepts that must be used.

That is why this thesis what it does is to optimize the C4R software made in 2011 that used obsolete software versions for the present and adapt the updated versions of Sumo, NetConvert and PolyConvert; leaving the possibility to continue updating the new versions that can be released from these software without losing functionality since new ways of reading, manipulating, and generating files for simulations were created.

Once the changes were made, performance tests were performed simulating real maps downloaded through the Open Street Maps server and the JMapView application, which is an application that can access and download updated maps from anywhere in the world.

Finally, the conclusions were drawn up and recommendations for future work and titling projects are mentioned.

KEYWORDS: Simulation, SUMO, Netconvert, Polyconvert, JMapView, Open Street Maps.

1. INTRODUCCIÓN

Actualmente, las redes de comunicaciones están disponibles en todo momento y casi en cualquier lugar. El ser humano quiere siempre estar conectado a internet [1] y acceder a todas las aplicaciones móviles que se encuentran en el mercado. Es por esto que tratando de encontrar nuevas maneras de comunicaciones fiables y dadas las ventajas que tienen las actuales redes de comunicaciones conocidas como Ad-Hoc nace el concepto de las VANETs (Vehicular Ad-Hoc Networks)[2].

Este tipo de redes son una nueva tecnología que está en pleno desarrollo ya que pueden cumplir con las expectativas de conectividad e información de los usuarios mientras están en movimiento. Además, las VANETs son redes tolerantes a fallos por cuanto forman una topología distribuida pero altamente dinámica; lo cual atrae la atención de los investigadores e industria para su desarrollo.

Dado que típicamente este tipo de redes involucrarán decenas o cientos de vehículos, un primer paso en su investigación y desarrollo es el uso de simuladores de generación de tráfico vehicular y protocolos de comunicaciones [3].

Por esta razón, para poder utilizar de manera más eficiente este tipo de redes se está trabajando mucho en las simulaciones ya que implementarlas en ambientes reales implicaría excesivo costo y desperdicio de recursos.

Partiendo de la premisa que para obtener resultados fiables se necesita tener variables de entrada lo más reales posibles y siendo uno de los principales problemas para el diseño de sistemas y aplicaciones en redes VANETs la falta de simulaciones apegadas a la realidad. Se pueden definir parámetros que servirían para obtener mejores resultados en las simulaciones antes mencionadas.

Estos parámetros son: trazado de vías, número de carriles, intersecciones y comportamiento del conductor. C4R (*Citymob for Roadmaps*) es una herramienta que facilita la obtención de trazas con un nivel de realismo adecuado, ya que incluye el uso de mapas libres descargados de OSMs (*Open Street Maps*) [4] y un motor de simulación de comportamiento vehicular potente como es SUMO (*Simulation of Urban MObility*)[4].

Sin embargo, a pesar de que C4R genera rutas realistas y que los modelos de movilidad usados por SUMO son ampliamente usados, C4R usa una versión ya obsoleta de SUMO para la actualidad. Esto repercute en el realismo del comportamiento de la conducción ya que los modelos son mejorados constantemente. Además, el uso de una versión antigua

de SUMO no permite explotar sus nuevas herramientas como son el control sobre el mapa, manipulación de intersecciones, rutas, volumen de tráfico, entre otras.

Esta tesis de grado propone actualizar y complementar la herramienta de software C4R para que cumpla con las expectativas de la comunidad de investigación de las VANETs. Esto quiere decir que el software a desarrollar realice simulaciones de tráfico vehicular con un alto nivel de realismo, usando para ello un conjunto de opciones disponibles en la herramienta de SUMO, Netconvert y Polyconvert que podrán incorporar la presencia de edificios, calcular atenuaciones entre vehículos; liberando de esta tarea al simulador de red y aumentando significativamente la confiabilidad en los resultados.

1.1 Pregunta de Investigación

El mejorar las herramientas a versiones actualizadas, permiten tener simulaciones y resultados más realistas por cuanto se cuenta con nuevas funcionalidades y se corrigen errores de las versiones anteriores.

1.2 Objetivo General

Rediseñar del Sistema C4R (*Citymob for Roadmaps*) con la implementación de nuevas opciones de las herramientas del motor de simulación de tráfico vehicular SUMO (0.30.0) para la obtención de simulaciones más realistas.

1.3 Objetivos Específicos

- Analizar las ventajas y desventajas del modelo de conexión Ad-hoc referente a las VANETs, los parámetros que intervienen en su aplicación a los modelos de movilidad y las herramientas complementarias como NETCONVERT y POLYCONVERT para la mejora en la manipulación de mapas de OSMs.
- Desarrollar los diagramas de flujo, de clases y de uso para los cambios del software C4R. Para su implementación.
- Analizar los resultados mediante pruebas de funcionamiento del aplicativo desarrollado.

1.4 Alcance

Esta tesis de grado adaptará la nueva versión de SUMO al aplicativo C4R que se encuentra desarrollado en Java, esto permitirá interactuar con el software y las herramientas en toda su capacidad y de una manera fácil para el usuario.

Incluir las principales opciones de NETCONVERT y POLYCONVERT mejorando la importación y manipulación de mapas obtenidos del sistema OSMS (Open Street Maps); con esto se podrá interactuar de una manera óptima y fácil con el archivo XML descargado.

Adicionalmente se podrá identificar el nombre de las vías y las intersecciones.

Los modelos de movilidad existentes no se modificarán, se podrán explorar y revisar las opciones de configuración para una fácil automatización. Cabe recalcar que todo será dentro de entorno de programación actual. Adicionalmente se menciona que el código de SUMO y los archivos de configuración no serán modificados.

1.5 Estructura del proyecto

La presente tesis de grado está compuesta de 4 capítulos, definidos de la siguiente manera.

El primer capítulo en el cual se hablará de la situación actual y de la motivación que se dio para el desarrollo de esta tesis, se mencionarán el objetivo general, los objetivos específicos para el desarrollo del aplicativo así como su alcance. De esto se podrá determinar los requerimientos necesarios y pasos a realizarse para su desarrollo.

Se incluirá además el marco teórico que será utilizado a lo largo del desarrollo del aplicativo, es en este capítulo en el cual se ha investigado los términos e información que se necesitan para cumplir los objetivos de la tesis.

El segundo capítulo está relacionado con la metodología y contiene todos los pasos necesarios para poder realizar la programación del mejoramiento del módulo de simulación, esto es diagrama de clases, diagrama de usos, diagrama de componentes, diagrama de navegación y secciones de código de los algoritmos utilizados para cumplir con los objetivos.

En el tercer capítulo se realizarán las pruebas correspondientes para evaluar los resultados y hacer una discusión de los estos.

El cuarto capítulo corresponde a las conclusiones y recomendaciones encontradas a lo largo del desarrollo de la tesis.

Además de los capítulos mencionados se encuentran tres anexos, el primer anexo se presentaran el código fuente del Software C4R mejorado escrito y desarrollado en NetBeans.

En el segundo anexo se presentara el código generado de las posibles rutas de las pruebas de simulación con el software C4R Mejorado para terminar con el tercer anexo, en el cual se mostrara las posibles rutas que genera el software C4R Original para un mismo mapa.

1.6 Marco Teórico

En esta sección que se desarrolla a continuación se estudiarán los conceptos necesarios para entender el desarrollo del proyecto.

Primero se parte de la definición de las redes Adhoc, redes VANETs, software de simulación principalmente enfocándose en sus partes y componentes.

Terminando con los modelos de movilidad y las estructuras de los archivos para su posterior manejo en las simulaciones.

1.6.1 Redes Adhoc y Redes VANETs [5]–[7]

Sin lugar a duda las redes Adhoc han tenido un crecimiento en esta última década ya que se puede armar una red sin la necesidad de una infraestructura determinada.

Lo más importante de las redes Adhoc es que se les puede añadir movilidad y de esto nace el nombre de las redes vehiculares VANETs (*Vehicular Ad-hoc Networks*), en donde se pone un nodo en cada vehículo creando una red móvil. [8]

Dentro de cada vehículo existe una unidad de comunicación a bordo llamada OBU (*On-Board Unit*) y una unidad de aplicación llamada AU (*Application Unit*).

La función principal de la OBU es el intercambio de mensajes a otros vehículos o puntos de acceso estacionarios ubicados alrededor de las carreteras denominados RSU (*Road-Side Unit*).

La función principal de las AU es mostrar la información al usuario final mediante interfaces al usuario final.

Se puede decir que en una VANET un vehículo se convierte en un nodo inalámbrico, y cada uno de estos se podría conectar con otro nodo que se encuentre de 100 a 300 metros a la redonda.

La idea de las VANETs es que la conexión depende de la intensidad de la señal, esto hace que un nodo pueda salir de la red, pero varios puedan ingresar.

Las VANETs se caracterizan por tener una topología altamente variable, movilidad absoluta e interacciones efímeras que podrían presentarse una sola vez.

Un dominio en una VANET son los elementos lógicos y físicos que operan entre sí para establecer las comunicaciones entre los nodos y las RSU. Estos dominios se clasifican de acuerdo con su funcionamiento.

1.6.2 Ventajas y desventajas de las redes Ad-hoc[9]

Como parte del estudio de las redes Ad-hoc se van a mencionar las ventajas y desventajas del uso de las redes Ad-hoc.

Ventajas

- Para establecer una conexión no se necesita de internet.
- Al crear una red Ad-hoc se puede compartir información entre todos los integrantes de la red formada.
- La información viaja directamente desde el emisor hacia el receptor sin pasar por ningún otro equipo intermedio.
- No se necesita de una administración central ni dispositivos de infraestructura para crear la red.
- Las configuraciones necesarias para crear la red son mínimas.

Desventajas

- Las tarjetas de interconexión de las redes Ad-hoc deben ser compatibles entre si para poder establecer la red.
- La configuración de la red debe realizarse cada vez que se quiere utilizar la red Ad-hoc, así sea haya creado anteriormente.

- Todos los equipos deben estar dentro del área de cobertura.
- Todos los nodos compiten por acceder al medio compartido inalámbrico, que puede derivarse en interferencias (colisiones), al querer transmitir mas de dos nodos a la vez.

Interpretación y análisis de las ventajas y desventajas de redes VANETs

Específicamente relacionado a las redes VANETs se pueden mencionar adicionalmente que al no necesitar una infraestructura, la dificultad recae en la coordinación ya que se carece de un coordinador que pueda asegurar un determinado nivel de calidad.

Para poder garantizar la correcta transmisión de información sea este audio, video, texto, etc. La confiabilidad en la entrega de información puede ser de extrema utilidad en este tipo de redes ya que pueden ser utilizados principalmente para video conferencias o por las autoridades de tránsito para monitorear el estado de un accidente.

Por otra parte debido a que muchos nodos están transmitiendo a la vez se pueden recibir números de secuencia de destino repetidos, en ese caso los números repetidos se eliminan (que se podría mencionar como ventaja), cosa que puede ser utilizado en redes altamente escalables. Una desventaja es el incremento del número de secuencia y el decremento de número de saltos, de una manera engañosa, que puede generar errores en la ruta.

Adicionalmente dentro de las ventajas esta la sencillez, robustez y el cambio parcial de actualización de enrutamiento con los nodos vecinos, ya que utiliza el método de descubrimiento de ruta mediante el protocolo DSDV (*Destination-Sequenced Distance Vector*), pero con la diferencia de que la información de cambio no se envía a todos los vecinos, sino que basándose en la información geográfica, la inundación de la red se la hará de manera controlada.

Protocolo DSDV (Destination-Sequenced Distance Vector routing)

El protocolo DSDV es esencialmente una modificación del algoritmo de encaminamiento Vector Distancia de Bellman-Ford empleado por en las redes fijas; este se lo puede encontrar en el protocolo RIP (*Routing Information Protocol*). En donde los nodos vecinos intercambian periódicamente sus tablas de enrutamiento completas con los vecinos para estimar la distancia a la que se encuentran los demás nodos no vecinos.

Las modificaciones introducidas por DSDV proporcionan esencialmente la obtención de rutas sin lazos mediante la introducción de números de secuencia para la determinación de las rutas más actuales.

Aunque DSDV sólo proporciona un camino para cada destino (no permite almacenar rutas secundarias o de soporte), siempre elige el camino más corto basándose en el número de saltos hacia este destino.

DSDV utiliza dos tipos de mensajes de actualización, uno más grande en tamaño (full-dump) y otro mucho más pequeño (incremental). Los mensajes incrementales pueden utilizarse para actualizaciones intermedias entre envíos periódicos (full-dump) de la tabla entera de enrutamiento.

1.6.3 Dominio en una VANET

Dominio en el vehículo

Este esta conformado por la OBU (*On-Board Unit*) y las AU (*Application Unit*) del nodo; forman una red de comunicación bidireccional dentro del vehículo y pueden conectarse alámbrica o inalámbricamente [6][10]

Dominio Ad hoc

En este dominio se hace referencia a la comunicación inalámbrica que se utiliza para enlazar dos o más nodos entre sí o los nodos con las RSU (*Roadside Unit*). Dicha comunicación puede establecerse mediante el estándar presentado por la IEEE denominado IEEE 802.11p (junto con sus variantes específicas para seguridad, *networking*, gestión de recursos y operaciones entre canales) también puede presentarse mediante otras tecnologías inalámbricas (WiFi, WiMAX, 3G, LTE) [11][12].

Dominio de infraestructura

Este dominio se encuentra formado por las redes de acceso y las redes de infraestructura que soporta el acceso a Internet (*backend*) que solicitan los nodos o las RSU. Esta comunicación puede realizarse utilizando tecnologías cableadas y/o inalámbricas [6]. Esto se lo puedo observar en la figura 1.1 que sigue a continuación.

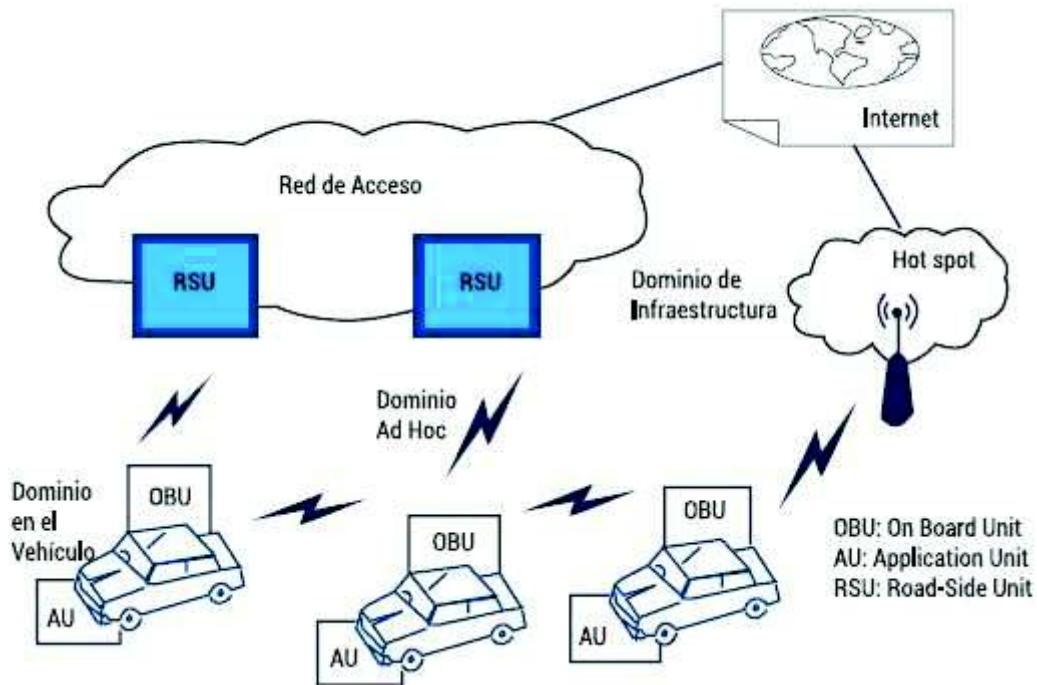


Figura 1.1. Dominios y componentes de una VANET [5]

1.6.4 Características de las redes Vehiculares [5] [13]

En las redes vehiculares se pueden definir dos tipos de comunicación, una de estas es la comunicación Inter Vehicular o vehículo a vehículo (V2V), en donde los vehículos cambian mensajes directamente y la comunicación vehículo a infraestructura (V2I), en las que el intercambio de mensajes se realiza con las RSU.

En conjunto las comunicaciones V2V y V2I se denominan V2X y tienen características como las que se mencionan a continuación.

Topología altamente dinámica

Llega a describir una topología específica para una red vehicular VANET es particularmente exigente debido a la misma naturaleza de los vehículos. Los nodos están en constante movimiento y una comunicación V2I o V2V puede darse en muy poco tiempo, dificultando la identificación de una topología para éstas redes [5].

Canales variables en tiempo y frecuencia

Debido a la velocidad de los vehículos, a los ambientes donde circulan y a los posibles obstáculos para la señal inalámbrica (edificios, árboles, etc.), la comunicación puede sufrir desvanecimientos en tiempo o frecuencia con mayor intensidad que otras redes móviles [6].

Autonomía

Hace referencia a la libertad que tiene cada nodo de la red para acceder al medio, transmitir, enrutar y recibir paquetes cuando lo requiera sin estar bajo la influencia de un control centralizado. De modo que las OBU y las RSU manejan dichas tareas de manera independiente [5][14].

Suministro de energía ilimitado

Los nodos no tienen restricciones en el consumo de energía, dado que la batería de los vehículos proporciona una cantidad suficiente para la operación de la OBU y/o AU [5].

Alta capacidad computacional

Las OBU en los nodos deben manejar, dependiendo de la aplicación, elevados flujos de tráfico de red con baja prioridad (en un *streaming* de video) o bajas tasas de datos con alta prioridad (aplicaciones de seguridad). Además controlan el enrutamiento y la conexión con las RSU, por lo que los dispositivos requieren capacidad computacional para realizar dichas tareas simultáneamente y sin latencias muy elevadas (en especial en aplicaciones críticas) [5].

1.6.5 Vehículos Inteligentes

Son vehículos experimentales los cuales son equipados con tarjetas con múltiples interfaces y sensores (OBU), con esto se podría tener una mejor sensibilidad en la conexión y por ende tener una mejor comunicación.

Una de las principales aplicaciones de este equipamiento son las notificaciones de accidentes de tránsito, congestiones vehiculares, de una manera más rápida y precisa[5].

1.6.6 Aplicaciones y modelos de movilidad [5]

Las VANETs dan la oportunidad de desarrollar aplicaciones para mejorar las condiciones del transporte y tráfico de los vehículos, esto mediante comunicaciones alternativas entre las V2X.

Entre las principales aplicaciones se pueden mencionar las de a continuación.

- Aplicaciones de seguridad vial.
- Disminución de la siniestralidad en las carreteras mediante la conexión de los vehículos y las RSU.
- Alerta de vehículos en sentido contrario
- Advertencias de cambios de carril
- Advertencias de conducciones peligrosas
- Notificación de señales de tránsito.
- Gestión de incidentes de tránsito
- Aplicaciones de eficiencia de tráfico.
- Gestión de tráfico.
- Monitoreo del tráfico.
- Aplicaciones de información y entretenimiento.
- Acceso a Internet.

1.6.7 JmapViewer

JmapViewer es un componente de java que permite fácilmente integrar y manipular los mapas del servidor de Open Street Maps, y estos poder ser embebidos desde una aplicación java. Provee varias herramientas que pueden ser administradas desde el panel principal.

1.6.8 Open Street Maps [15]

OSM (Open Street Maps) es un proyecto colaborativo para crear mapas libres y editables. Los mapas se crean utilizando información geográfica capturada con dispositivos GPS móviles, mediante fotografías y otras fuentes libres.

Las imágenes de cartografía se almacenan en base de datos y se distribuye bajo licencia libre y abierta de base de datos conocida como ODbL (*Open Database Licence*).

Open Street Maps tiene una gran acogida, ya que cada año se registran varios nuevos usuarios, en desarrollo de esta tesis de grado se tienen registrados 4.604401 [7].

De esto se puede decir que existe un gran interés en el desarrollo de esta plataforma y la forma de utilización de esta.

Es importante esta información ya que los usuarios registrados y validados por la plataforma pueden subir sus trazas desde un dispositivo GPS, y corregir datos vectoriales mediante herramientas de edición creadas por la comunidad de Open Street Maps.

Según estadísticas de la misma fuente cada semana se añaden alrededor de 90000Km de carreteras, lo que lo hace una fuente casi real para realizar simulaciones.

Formato de Datos [15]

Open Street Maps utiliza una estructura de datos topológica, donde los datos se almacenan en el datum WGS84 (*World Geodetic System*) que es un sistema de coordenadas geográficas mundial que permite localizar un punto de la tierra dando tres datos de entrada.

Este sistema tiene una alta confiabilidad por cuanto se pueden tener errores de 2cm y es lo que actualmente se usa en el posicionamiento global o GPS.

Los datos primitivos o elementos básicos en la cartografía de Open Street Maps son los siguientes:

Nodos (*nodes*): Son puntos específicos que recogen una ubicación específica dada.

Vías (*ways*): Son una lista ordenada de nodos que representa un polígono.

Relaciones (*relations*): Son grupos de nodos, vías u otras relaciones que se pueden asignar determinadas propiedades comunes.

Etiquetas (tags): Son datos que se asignan a nodos, caminos o relaciones que constan de una clave (key) y de un valor (value). Por ejemplo highway=trunk, define una vía como carretera como troncal.

Del proyecto de OSM, se pueden definir mapas de carreteras, mapas de senderismo, mapas de vías de bicicletas, mapas náuticos, etc.

La cartografía de Open Street Maps utiliza datos en dos dimensiones (no toma en cuenta la altura), pero existe a la par otro proyecto llamado Misión topográfica Radar Shuttle (SRTM) para crear mapas topográficos con datos de altura.

El formato que utiliza Open Street Maps es el OSM - XML, tal como se muestra en la Figura 1.2

```

<?xml version="1.0" encoding="UTF-8"
<osm version="0.6" generator="OSMapi 0.6.0 (9712 thorn-01.openstreetmap.org)" copyright="OpenStreetMap and contributors"
attribution="http://www.openstreetmap.org/copyright" license="http://openstreetmap.org/licenses/odhl/1.0/"
<boundbox minlat="-0.1498300" minlon="-78.4976100" maxlat="-0.1472000" maxlon="-78.4948600"/>
<node id="264553194" visible="true" version="3" changeset="32710520" timestamp="2015-01-18T00:46:18Z" user="F99M" uid="
1426525" lat="-0.1497794" lon="-78.4973901"/>
<node id="264553195" visible="true" version="3" changeset="32710520" timestamp="2015-01-18T00:46:18Z" user="F99M" uid="
1426525" lat="-0.1706075" lon="-78.4976387"/>
<node id="264553196" visible="true" version="4" changeset="39546047" timestamp="2014-05-24T22:20:47Z" user="GeoCrazy" uid="
3693002" lat="-0.1710402" lon="-78.4975777"/>
<node id="264558123" visible="true" version="5" changeset="54327664" timestamp="2017-12-04T12:27:57Z" user="lorandr_talanav"
uid="4973384" lat="-0.1703019" lon="-78.4983017">
<tag k="highway" v="turning circle"/>
</node>
<node id="264558122" visible="true" version="3" changeset="39546047" timestamp="2014-05-24T22:20:47Z" user="GeoCrazy" uid="
3693002" lat="-0.1493303" lon="-78.4981966"/>
<node id="264558125" visible="true" version="3" changeset="39546047" timestamp="2014-05-24T22:20:47Z" user="GeoCrazy" uid="
3693002" lat="-0.1499047" lon="-78.4980303"/>
<node id="264558139" visible="true" version="2" changeset="11550459" timestamp="2012-05-09T18:23:00Z" user="WurstFleiss" uid="
678132" lat="-0.1494025" lon="-78.4978092"/>
<node id="264558142" visible="true" version="2" changeset="11550459" timestamp="2012-05-09T18:23:00Z" user="WurstFleiss" uid="
678132" lat="-0.1497244" lon="-78.4974819"/>
<node id="264558145" visible="true" version="4" changeset="54327664" timestamp="2017-12-04T12:27:57Z" user="lorandr_talanav"
uid="4973384" lat="-0.1700165" lon="-78.4960142">
<tag k="highway" v="turning circle"/>
</node>
<node id="264558151" visible="true" version="3" changeset="40193350" timestamp="2014-06-21T22:55:50Z" user="Baconcrisp" uid="
3778777" lat="-0.1489377" lon="-78.4964223"/>
<node id="264558161" visible="true" version="7" changeset="40208390" timestamp="2014-06-22T15:32:49Z" user="Baconcrisp" uid="
3778777" lat="-0.1484250" lon="-78.4967727"/>
<node id="264558144" visible="true" version="4" changeset="40193350" timestamp="2014-06-21T22:55:50Z" user="Baconcrisp" uid="
3778777" lat="-0.1479128" lon="-78.4969040"/>

```

Figura 1.2. Sección de archivo OSM.xml descargado por JMapViewr

Para poder adaptar la nueva versión de SUMO y hacerla más práctica para el usuario final, se tiene que entender cómo se distribuyen los datos en los archivos XML descargados y generados, en el capítulo de desarrollo y programación se detallará como se elaboraron los algoritmos de lectura y escritura.

1.6.9 NETCONVERT

NETCONVERT es una herramienta de SUMO basado en el lenguaje de programación C++, que es usada para importar redes de carreteras digitales de diferentes fuentes, para ser usado en otras herramientas de la misma plataforma.

El principal propósito es la importación y conversión de las redes de carreteras, funciona en sistema Linux o Windows. Para el propósito de esta tesis se lo realizará en sistema Windows.

Ocupa líneas de comandos y genera redes de carreteras con diferentes parámetros de salida.

Esta herramienta por ejemplo puede importar desde diferentes tipos de formatos como son:

- OpenStreetMap (*.osm.xml).
- VISUM, incluyendo figuras y demandas.
- OpenDRIVE
- MATsim
- SUMO (*.net.xml).

El uso de NETCONVERT es de manera sencilla, en la línea de comandos, por ejemplo, para importar una red de carreteras desde *Open Street Maps* se tendría que utilizar el siguiente código [4]

```
netconvert --osm mi_osm_net.xml
```

Para poder definir un nombre al archivo generado de este comando se tendría que aumentar el parámetro `-o`, por ejemplo, de esta manera [4]:

```
netconvert --osm mi_osm_net.xml -o mi_sumo_net.net.xml
```

Existe una serie de parámetros adicionales que se tiene para la opción de NETCONVERT, en este caso no se mencionarán por cuanto no son necesarios para este trabajo.

1.6.10 POLYCONVERT [4][11]

Es también una herramienta programada en C++ de SUMO que importa formas geométricas (polígonos o puntos de interés) de diferentes fuentes y las convierte en una representación que puede visualizarse en la interfaz gráfica de SUMO.

Igual que la herramienta NETCONVERT funciona mediante línea de comandos, donde se tiene una entrada obligatoria y consta de varios parámetros de salida de acuerdo con lo que el usuario necesite.

POLYCONVERT puede importar formas de diferentes tipos de archivos, normalmente para importar datos de un cierto tipo, el tipo es usado como opción y el valor indica la posición del archivo, de esto se puede indicar el siguiente ejemplo:

```
polyconvert --osm mired.net -o converted.poi.xml
```

POLYCONVERT es capaz de aplicar diferentes atributos a las formas importadas en dependencia de su "tipo". No todos los formatos importados tienen información de tipo.

Cuando se usan archivos de formas, por ejemplo, todas las instancias de un tipo de artefacto normalmente se almacenan en un archivo de formas distintas [8].

Uno de los principales usos de POLYCONVERT es aplicar proyecciones, sobre las formas leídas, normalmente se requiere que las formas se alineen de acuerdo a la red vial previamente importada usando el comando `--net-file <FILE>`.

Pero también es posible usar una proyección diferente con el código `--use-projection`.

Cuando las formas cubren un área mucho más grande que la red, se tienen que reducir los datos, para esto se tiene la opción *Prune* que obliga a POLYCONVERT a reducir los datos a un límite determinado. Las Formas que están completamente fuera de este límite se descartarán.

Al importar formas para las que no se han otorgado atributos dependientes del tipo, se utilizan los valores predeterminados que se pueden cambiar en la línea de comandos [4]. Todo estos parámetros serán definidos en el archivo `typemap.xml` que será generado dependiendo de los parámetros de entrada que se quisieran visualizar.

1.6.11 SUMO (Simulation of Urban Mobility)

Es un paquete de simulación de tráfico de código abierto, diseñado para poder interpretar grandes redes de carreteras o vías, es un simulador de gran importancia en las telecomunicaciones que tiene las características que se mencionan a continuación.

- Interpretación de reglas de conducción, semáforos.

- Administración de redes con más de 10000 nodos.
- Alta velocidad de ejecución.
- Interoperabilidad con otras aplicaciones en tiempo real.
- Importación de redes.
- Importación de archivos XML de cualquier simulador de redes vehiculares.
- Valores faltantes se determinan de manera heurística.
- Enrutamiento.
- Rutas microscópicas, cada vehículo tiene su propia ruta.
- Diferentes algoritmos dinámicos de asignación de usuarios
- Alta portabilidad.
- Bibliotecas y código C++ se usan solamente.
- Puede coexistir entre Windows y Linux.

SUMO por defecto tiene una opción para simular aleatoriamente el paso de los vehículos a través de una red vial dada, pero también es posible definir manualmente el archivo de demanda.

Es importante entender que un vehículo en SUMO consta de tres partes principalmente.

- Un tipo de vehículo que describe las propiedades físicas del mismo.
- Una ruta que tomará el vehículo.
- El vehículo propiamente dicho.

Rutas y vehículos pueden presentarse en el mismo lugar, el movimiento de estos también puede ser modelado de acuerdo con el usuario, si no existen parámetros de entrada relacionados a esto, se utilizarán los definidos por defecto, a continuación se describen las principales ventajas de la nueva versión de SUMO.

Definición de Rutas

Un vehículo puede transitar por cualquier calle de la red vial extraída del api de Open Street Maps, SUMO por defecto asignará a un vehículo una ruta aleatoria, pero si el usuario quiere hacer pruebas más reales, se podrán crear rutas que serán asignadas a determinados vehículos.

Estas rutas deben constar del inicio, medio y final, en la siguiente sección de código se crea una ruta donde el vehículo conducirá a lo largo las calles "inicio", "medio", "fin", y tan pronto como se acerque al borde "final" se eliminará de la simulación.

Definición de Vehículos

Como es lógico, un vehículo va a tener su aceleración, desaceleración, velocidad máxima y un tipo de vehículo, en el ejemplo de a continuación mediante codificación xml, se definirá un vehículo con una ruta que solo este posea.

```
<routes>
  <vType id="type1" accel="0.8" decel="4.5" sigma="0.5" length="5"
  maxSpeed="70"/>

  <vehicle id="0" type="type1" depart="0" color="1,0,0">
    <route edges="beg middle end rend"/>
  </vehicle>
</routes>
```

Figura 1.3. Código para definición de vehículo en una ruta.

Aprovechando la interfaz gráfica de SUMO, el usuario puede definir un vehículo de color rojo color = 1,0,0) de tipo "tipo 1" llamado "0" que comienza en el tiempo 0 su movimiento.

Este vehículo tiene su propia ruta interna que no se comparte con otros vehículos. También es posible definir dos vehículos usando la misma ruta. En este caso, la ruta debe ser definida antes de ser utilizada por los vehículos.

Además, la ruta debe nombrarse dándole un id. Los vehículos que usan la ruta lo refieren utilizando el atributo "ruta". En código esto se vería de la siguiente manera:

```
<routes>
  <vType id="type1" accel="0.8" decel="4.5" sigma="0.5" length="5"
  maxSpeed="70"/>

  <route id="route0" color="1,1,0" edges="inicio medio fin final"/>

  <vehicle id="0" type="type1" route="route0" depart="0" color="1,0,0"/>
```

```

<vehicle id="1" type="type1" route="route0" depart="0" color="0,1,0"/>
</routes>

```

Figura 1.4. Código para definición de una ruta para un vehículo con características.

Definición de tipos de vehículos

En SUMO un vehículo puede ser definido por el parámetro vType como se muestra en la siguiente sección de código, en donde el tipo de vehículo será “tipo1” con una aceleración de “2.6”, una desaceleración de “4.5” y una máxima velocidad de “70”.

```

<routes>
  <vType id="tipo1" accel="2.6" decel="4.5" sigma="0.5" length="5"
  maxSpeed="70"/>
</routes>

```

Figura 1.5. Código para definición de características físicas de un vehículo

Definiendo los tipos de vehículos, se pueden construir más vehículos del “tipo1”, en la siguiente sección de código se crean el “veh1” y “veh2” con las mismas características del tipo “tipo1”. Además el “veh1” tomará la ruta “edge1 edge2 edge3” y el “veh2” la ruta “edge4 edge5 edge6”.

```

<routes>
  <vType id="tipo1" accel="2.6" decel="4.5" sigma="0.5" length="5"
  maxSpeed="70"/>
  <vehicle id="veh1" type="tipo1" depart="0">
    <route edges="edge1 edge2 edge3"/>
  </vehicle>
  <vehicle id="veh2" type="tipo1" depart="0">
    <route edges="edge4 edge5 edge6"/>
  </vehicle>
</routes>

```

Figura 1.6. Código para definición de varios vehículos y varias rutas

Todo lo mencionado en estas definiciones anteriormente mencionadas se refiere a características meramente físicas, como color, velocidad máxima, aceleración, desaceleración.

Se pueden definir más parámetros a los tipos de vehículos, estos se pueden observar en la tabla 1.1 que se muestra a continuación.

Tabla 1.1. Opciones adicionales en definición de vehículos

Nombre Atributo	Descripción
Id	El nombre del vehículo
Type	El Tipo del vehículo
Route	La id de la Ruta a tomar el vehículo
Color	El Color del Vehículo
Depart	El tiempo en el que el vehiculo entra a la red vial
DepartLane	El carril en el que se insertará el vehículo
DepartPos	La posición en la que el vehículo ingresará a la red
DepartSpeed	La velocidad con que el vehículo ingresará a la red
ArrivalLane	El carril en el cual el vehículo deberá abandonar la red
ArrivalPos	La posición en la que el vehículo abandonará la red
ArrivalSpeed	La velocidad con la que el vehículo debe abandonar la red
PersonNumber	La cantidad de asientos ocupados cuando se inserta el vehículo
ContainerNumber	La cantidad de contenedores ocupados cuando se inserta el vehículo
Via	Lista de bordes intermedios que se pasarán en el reencaminamiento
DepartPosLat	La posición lateral en la línea de salida en la cual el vehículo ingresará a la red
ArrivalPosLat	La posición lateral en la línea de llegada a la que llegará el vehículo

Para tener aún simulaciones más realistas, se tienen que tomar en cuenta los modelos de movilidad de los vehículos.

1.6.12 Definición de Modelos de Movilidad de Vehículos (Car-Following Models)[13][4]

Un modelo de movilidad se puede describir como un conjunto de reglas que pueden describir el comportamiento de los vehículos en movimiento. Un modelo de movilidad también puede definirse como una pieza más en un sistema que pretende interconectar varios vehículos y puede describir cómo tiene que desplazarse un vehículo y qué decisiones tiene que tomar.

Existen diversos modelos de movilidad muchas veces eligen entre complejidad y detalle, es decir, estos se eligen entre aproximarse a la realidad de una forma relativamente simple, analizando solamente los principales factores, o aproximarse a la realidad con el máximo de detalle posible, donde se estudian todos los factores que pueden influir.

En cualquier caso es importante que los modelos de movilidad sean fieles a la realidad. Para conseguirlo es necesario tener en cuenta aspectos básicos que influyen en la movilidad.

La disminución de la velocidad favorece que haya menos cambios en el enrutamiento de datos de la red. Pero por otra parte los *clusters* de vehículos pueden afectar negativamente al funcionamiento de la red por suponer un aumento en la contención del canal.

Movimiento interdependiente de los vehículos[1]

A pesar de que cada vehículo puede tomar sus propias decisiones, éste se mueve influido por los vehículos que le rodean.

Velocidad media

La velocidad de un vehículo establece cómo de rápido cambia su posición, lo que determina a la vez cómo cambia la topología de red.

La media de velocidad varía en función de los límites de velocidad establecidos las características y el estado de la calzada, la aceleración, deceleración, y la topología de las calles. Del mismo modo, si hay pocas intersecciones los vehículos aceleran a velocidades mayores que si hay muchas intersecciones

Para poder obtener resultados más reales se necesitan involucrar la mayor cantidad de variables que intervienen en el movimiento y esto se lo hace dando al usuario la posibilidad de seleccionar el modelo de movilidad que quiere utilizar en la simulación.

Tipos de Modelos de movilidad (Clasificación genérica)

Modelos Sintéticos

Son los obtenidos por la teoría matemática por lo que los resultados obtenidos van a depender de las influencias recibidas, entre los principales se distinguen:

- Modelos estocásticos o aleatorios.
- Modelos de flujo de tráfico.
- Modelos de vehículo seguidor.
- Modelos de cola.
- Modelo de comportamiento.

Modelos basados en datos

Estos se basan en su mayor parte en la información recopilada en un periodo de prueba de forma macroscópica y están directamente relacionadas a estadísticas con el fin de encontrar comportamiento pseudo aleatorio [16].

Modelos basados en trazas

Se pretende extraer trazas reales a partir de bases de datos del mundo real, es decir hallar modelos matemáticas a base de estos [16].

Modelos basados en simuladores de tráfico

Estos modelos son los obtenidos del refinado de los modelos sintéticos validados por las trazas reales y datos de comportamiento que han desembocado en simulaciones de tráfico reales[16].

En una VANET los modelos de movilidad permiten describir el movimiento de los nodos dentro de un escenario, en este caso como se va a utilizar mapas reales se debe considerar 4 tipos de modelos de tráfico (macroscópico, microscópico, submicroscópico, y mesoscópico).

De esto los macroscópicos se enfocan en la identidad básica del flujo vehicular, por lo que incorporan aproximaciones en términos de velocidad, flujo y densidad de tráfico, los microscópicos pueden simular el movimiento de cada vehículo independiente en la vía (características físicas), los submicroscópicos similares a los microscópicos pero extiende un enfoque dividiendo en subestructuras (velocidad del motor, cambio de marchas, etc).

Finalmente están las simulaciones mesoscópicas que se encuentran entre las simulaciones microscópicas y macroscópicas, éstas no consideran la velocidad dinámica individual del vehículo sino que asumen un valor medio a lo largo del viaje en un determinado tramo.

SUMO ocupa modelos de movilidad microscópicos para las simulaciones de tráfico, debido a esto se analizarán los modelos propuestos por S. Krauß, P. Wagner, Kerner y Treiber.

Modelo Krauss [17], [18]

Es un modelo de movilidad microscópico propuesto por Stefan Krauß, el cual en los cálculos incluye propiedades del flujo de tráfico que se caracteriza en el hecho de que las interacciones entre los vehículos son motivadas por la intención de no colisionar unos a otros.

Es por eso que este modelo permite representar las variaciones de velocidad producidas por la dependencia correspondiente a la mínima distancia de frenado que deben tener los conductores respecto al vehículo que les precede en la vía y de su velocidad de desplazamiento.

Modelo Wagner [17], [19]

Este modelo fue desarrollado por Peter Wagner, con el fin de introducir dos características importantes de la conducción humana: la primera las personas suelen planificar el evento futuro mientras conducen, y la segunda que el tipo de control que ejercen sobre sus vehículos, no es continuo pero si discreto en el tiempo, es decir sólo actúan en determinados momentos denominados puntos de acción, los cuales son considerados como fenómenos aleatorios; pues suceden con mayor o menor motivo en función del entorno y la situación del conductor.

Modelo Kerner [1]

Conocido también como teoría del tráfico trifásico porque divide el tráfico vehicular en tres fases para modelarlo: flujo libre, flujo sincronizado y amplio congestionamiento. A continuación la explicación de cada uno de estos tres estados:

- **Flujo libre:** En este estado los vehículos pueden circular sin problemas de congestión.
- **Flujo sincronizado:** El término “sincronizado” se refiere a la tendencia de sincronización, de las velocidades de los vehículos en la carretera; debido a la baja probabilidad de paso que hay en el tráfico congestionado. En el flujo sincronizado los vehículos tienden a agruparse en hileras durante su desplazamiento pero pasado el límite, se crea un cuello de botella; que impide la libre circulación.

- **Amplio congestionamiento:** Es un caso de tráfico congestionado; que se puede dar cuando la densidad de tráfico es extremadamente alta y la velocidad a la que circulan los vehículos es casi nula.

Modelo de Conducción Inteligente - IDM

IDM es un modelo determinista, en el cual la aceleración de un vehículo depende de la suya propia, de la de los vehículos que le rodean y de la distancia mantenida hacia el vehículo precedente. Con IDM es posible simular además de los aspectos relacionados con el vehículo y su entorno, el comportamiento de los conductores para lo cual supone tres tipos diferentes: agresivo, normal y tímido; a más de permitir diferenciar entre conductores de autos pequeños y camiones.

Sin embargo, los valores pueden variar de acuerdo a los límites de velocidad establecidos por la ley de transporte y seguridad vial de un país o región en específico.

1.6.13 Estudio y análisis de los modelos de Movilidad para su uso

Una vez definidos los términos necesarios para entender que son los modelos de movilidad y mencionados también los parámetros que intervienen e influyen en la trayectoria de un vehículo a lo largo de una ruta, se hará un estudio de los modelos de movilidad y del comportamiento espacial de los vehículos durante un periodo de tiempo, conceptos necesarios para poderlos utilizar en la realización de las simulaciones.

Un elemento muy importante para poder elegir el correcto modelo de movilidad para realizar una simulación es que sea compatible con las características al igual que el ambiente en donde se quiera realizar dicha simulación.

Otro punto importante a entender es que el comportamiento espacial de los vehículos depende de varios factores que se enlistan a continuación.

Es por eso que de estos conceptos se van a estudiar los principales parámetros para según el criterio adecuado sean ocupados en la realización de las simulaciones que se requieran.

Tipo de vehículo que se requiere simular

Los vehículos que circulan a lo largo de un espacio designado para su movimiento pueden ser de distintos tamaños, de distinto tipo de motor, de distinta tecnología por lo tanto se debería seleccionar un modelo que abarque este tipo de condiciones.

Influencia vehicular a los alrededores del vehículo

Este parámetro se encuentra considerado en el modelo IDM Modelo de Conducción Inteligente, es muy importante ya que en la realidad el movimiento espacial de un determinado vehículo depende también de la influencia que se puede presentar en un instante determinado por determinados vehículos en su alrededor.

Influencia humana en el manejo del vehículo

Este concepto viene del Modelo Wagner en el cual las personas pueden planificar un evento mientras conducen, y que el tipo de control que ejercen sobre sus vehículos, no es continuo sino discreto en el tiempo. Cosa que en la realidad es así.

Estos conceptos serán utilizados con bastante criterio en la creación de las rutas y sobre todo cuando se seleccione un nuevo vehículo que debe circular a lo largo de la ruta determinada.

Las reglas de conducción

En estas reglas de conducción intervienen varios parámetros que el vehículo debe cumplir, si se cumplen estas reglas se podría decir que la simulación esta apegada a la realidad.

- **Trazado de vías:** Los vehículos tienen que moverse obligatoriamente dentro de las calles, por lo que la movilidad queda limitada a su trazado.
- **Tamaño de la manzana:** Una manzana puede ser considerada como la menor área rodeada por calles. El tamaño de la manzana determina el número de intersecciones en un área y por tanto establece la frecuencia con la que un vehículo tiene que parar en una intersección y la distancia a la que un vehículo vecino es capaz de detectar la transmisión de otro vehículo.

- **Mecanismos de control de tráfico:** Las formas más comunes de controlar las intersecciones son las señales de stop y los semáforos. Estos mecanismos crean clusters y colas de vehículos en las intersecciones. Como es lógico estas colas provocan una reducción de la media de la velocidad de los vehículos.

Es muy importante mencionar que el modelo elegido necesariamente debe acercarse a la realidad del ambiente de simulación pero es también obligatorio el ser compatible con la herramienta SUMO que es la encargada de integrar estos conceptos.

SUMO en su versión 0.30, al ser una herramienta altamente eficiente para realizar simulaciones realistas tiene varias opciones para poder aumentar la veracidad en los resultados de las simulaciones.

En la versión 0.30 se puede ocupar los modelos de movilidad mejorados que se quiera analizar, solo se los debe incluir como un parámetro adicional en el archivo a ejecutarse en el *command prompt* de Windows.

De esto se puede asignar a los vehículos que transitan por una red vial (extraída de OSM) un modelo de movilidad, sea aleatoria, o vehículo por vehículo.

Las posibles opciones de modelos de movimiento de vehículos en el simulador de SUMO son las que se presentan en la tabla 1.2.

Tabla 1.2 Modelos de movilidad

Atributo en Línea Comandos	Modelo de Movilidad
carFollowing-Krauss	Krauss
carFollowing-KraussOrig1	KraussOrig1
carFollowing-PWagner2009	PWagner2009
carFollowing-BKerner	BKerner
carFollowing-IDM	IDM
carFollowing-IDMM	IDMM
carFollowing-KraussPS	KraussPS
carFollowing-KraussAB	KraussAB
carFollowing-SmartSK	SmartSK
carFollowing-Wiedemann	Wiedemann
carFollowing-Daniel1	Daniel1

En la siguiente sección de código se define un vehículo de “tipo1” que además de sus parámetros antes mencionados como velocidad, aceleración, desaceleración, ocupa el modelo de movilidad “Krauss”.

Cabe mencionar que el modelo “Krauss” es el que está definido por defecto si es que no se lo define.

```
<routes>
  <vType id="tipo1" length="5" maxSpeed="70" carFollowModel="Krauss"
  accel="2.6" decel="4.5" sigma="0.5"/>
</routes>
```

Figura 1.7. Código para definición de vehículo con modelo de movilidad “Krauss”

Definición de Rutas, nodos y referencias

Como se puede observar, una de las mejoras que se realiza en esta tesis de grado, es la personalización de diferentes componentes que se ocupan en la elaboración de simulaciones.

Al analizar los archivos de mapas descargados del API de Open Street Maps, se pudo determinar que el archivo xml obtenido tiene información de las vías como es:

- Nombre de la vía.
- Tipo de vía.
- Ubicación de la vía.
- Usuario de OSM que actualizo la información.
- Referencias a nodos.

De esto, se pretende construir un algoritmo que permita al usuario de una manera fácil, rápida y gráfica escoger la vía por donde quiere que el vehículo se mueva a través del mapa descargado de Open Street Maps.

2. METODOLOGIA

La metodología ocupada para la elaboración de esta tesis es mixta, primera se hará un proceso lógico para la elaboración de las nuevas herramientas siguiendo modelos de programación para después realizar experimentos para comprobar el correcto funcionamiento y comprobar los resultados obtenidos, además se basará en la metodología expuesta por Kendall y Kendall de donde se tendrá que identificar el problema, señalar e identificar las oportunidades y objetivos para terminar con la documentación del software, Informes, entregables y pruebas e implementación del mismo.

2.1 Identificación del problema, oportunidades y objetivos

Como se mencionó en el primer capítulo de este proyecto de tesis una de las principales desventajas del software C4R es que se ocupa como base la versión de SUMO 0.11, que es una versión ya obsoleta por cuanto tiene varias opciones que en la actual versión no se usan o ya fueron eliminadas.

Otra de las desventajas es que la manera de manipular y generar los archivos de rutas es totalmente diferente, por ejemplo, el actual C4R se encuentra funcionando con la versión de SUMO 0.11 maneja para la discriminación de rutas el uso etiquetas “*EDGES*” en los archivos xml, cosa que en la versión actual de SUMO 0.30 ya no se utiliza, actualmente (versión de SUMO 0.30) los archivos generados tienen etiquetas de nodos *NODES* y referencias *REFERENCE* a los nodos en un archivo XML.

Este cambio encontrado en los archivos generados de la nueva versión de SUMO hace que las simulaciones sean más reales, ya que se puede considerar una mayor cantidad de parámetros.

2.2 Objetivos y oportunidades para el desarrollo

Aprovechando las nuevas ventajas de la versión de SUMO 0.30 se incluyó las siguientes mejoras a realizarse:

- Crear unas opciones para la utilización de POLYCONVERT, mediante la cual se podrá escoger cuales parámetros deberán intervenir en el proceso.
- Implementar un mecanismo mediante el cual el usuario pueda crear un tipo de Vehículo, con su respectivo modelo de movilidad, clase, color, características de movimiento entre otras.
- Permitir la personalización en la construcción de rutas que va a tomar un número “n” de vehículos determinados.

De estas mejoras que se van a realizar en el rediseño del software, se explicarán los diagramas de casos de uso, diagrama de clases y funciones que se necesitan crear y definir.

Para establecer un orden jerárquico y ordenado se elaborará todo el proceso necesario basándose en el principio de la Ingeniería de Software UML (*Unified Modeling Language*).

2.3 Análisis y necesidades de C4R

Al ocupar la herramienta actual de C4R en toda su funcionalidad se pudo determinar las necesidades y falencias que tiene la herramienta actual de C4R, de este análisis se podrán mencionar las nuevas funcionalidades que debería tener la herramienta nueva en comparación con la herramienta anterior.

Funcionalidades de C4R en su versión original

La herramienta de C4R en su versión original cuenta entre las características principales las que se listan a continuación.

- Selección de mapas mediante Open Street Maps con una limitación de 1 km².
- Descarga de mapas mediante algoritmos programados.
- Creación de simulaciones aleatorias con la inserción de varios parámetros, mediante una manera visual.
- Creación de simulaciones personalizadas, mediante algunas ventanas con formularios que se despliegan según el uso.

- Creación de rutas mediante formularios gráficos.
- Creación de tipos de vehículos mediante formularios gráficos.
- Creación de archivos para su uso en programas que realizan la simulación específicamente.

De cada una de estas características que se encuentran listadas se analizarán las partes negativas y se mencionarán las posibles soluciones que se deberían incluir en la nueva versión de este software.

Con esta información en las próximas fases de la elaboración del proyecto se podrán determinar los requerimientos, se podrán elaborar las clases, los diagramas para terminar con la programación de las funciones y la programación de los algoritmos necesarios.

Selección de mapas mediante Open Street Maps

Este proceso se encuentra funcionando correctamente y no se detecta muchas dificultades, lo que se recomienda y se realizará en la nueva versión de C4R será el no limitar a 1km² si no se lo limitará a 3km² para poder realizar simulaciones en ambientes mucho mas grandes, un punto importante es que el api de conexión y el formato de datos es diferente.

Descarga de mapas mediante algoritmos programados

Este proceso es específico de la versión de C4R original, si bien el software mejorado tendrá que realizar el mismo proceso, este será totalmente diferente ya que la estructura de los archivos y el manejo de éstos va a ser diferente.

Creación de simulaciones aleatorias

Para la creación de simulaciones aleatorias el proceso que se tiene en el programa en su versión actual de C4R es difícil en lo relacionado al ingreso de información, lo que se tratará de hacer en la versión mejorada de C4R es programar un formulario gráfico para que el usuario al realizar una simulación no necesite ingresar información no necesaria.

Creación de simulaciones personalizadas

Se puede mencionar que en la versión original de C4R se puede hacer pero es una tarea difícil ya que la manera de presentar los archivos y datos es incompleta, en este punto es el que se tratará de poner mayor énfasis en el desarrollo del aplicativo y es el que hizo emprender este proyecto de titulación, lo que se realizará en la nueva versión mejorada de C4R es que el usuario pueda ingresar y crear de una manera gráfica todos los parámetros que se puedan necesitar, estos parámetros son los que se mencionarán en los siguientes puntos.

Creación de rutas mediante formularios gráficos

La creación de rutas en el software C4R en su versión original existe pero el proceso no es amigable para el usuario ya que al cargar las rutas en el “jCombobox” del programa C4R en su versión original, éstas solo son representadas por su identificativo (un valor numérico), cosa que hace difícil o imposible determinar a que calle representa el valor numérico seleccionado.

Adicionalmente en esta misma opción de elaboración de rutas, no se tienen opciones fundamentales como poder visualizar la velocidad permitida, nombre de la calle, tipo de la calle y hasta poder visualizar en el mapa que calle es la que se esta seleccionando.

Creación de tipos de vehículos mediante formularios gráficos

En esta opción que consta en el programa C4R en su versión original se puede crear una tipo de vehículo para toda la simulación, es decir no se tiene la opción de crear vehículos indistintamente para el proyecto. Además actualmente los vehículos en las simulaciones pueden tener mayor cantidad de atributos que ayudan a mejorar las simulaciones estas opciones pueden ser:

- Tipo de vehículo.
- Tipo de modelo de movilidad.
- Aceleración.
- Velocidad máxima.
- Color del vehículo.

Entonces se puede concluir que una necesidad importante sería poder incluir de alguna manera los atributos de los vehículos antes mencionados para las simulaciones.

Otro punto importante es que estas opciones deberían crearse mediante una interfaz fácil de utilizar y debería incorporarse dentro de una misma pantalla para poder asignar la ruta y el tipo de vehículo que se mueva a través de la ruta creada.

Creación de archivos para su uso en programas

Por cuanto el programa en su versión original data del año 2011, el software ocupado en esta versión es obsoleto, es decir las funcionalidades de los programas son limitadas y hasta el manejo de la información es diferente. Por lo que se tendría que implementar una nueva manera de interpretar los datos y ocupar las nuevas funcionalidades del software especialmente lo que es SUMO, PolyConvert y NetConvert.

Del análisis mencionado anteriormente se determina que el programa C4R puede mejorarse de la siguiente manera y cumpliendo las siguientes actividades.

- Establecer de una manera gráfica las opciones de selección de rutas con los nombres y la velocidad.
- Realizar una vista previa de la calle seleccionada en el programa, para poder determinar los posibles nuevos saltos a otras calles.
- Mediante algoritmos poder presentar solamente las calles que tienen conexión y pueden acceder (determinando el sentido es decir si es una vía o doble vía).
- Desarrollar un formulario mediante el cual se pueda crear los diferentes tipos de vehículos considerando estas características.
 - Tipo de vehiculo (taxi, camión, motocicleta, etc.)
 - Tipo de modelo de movilidad (todos los tipos de movilidad que permita SUMO)
 - Aceleración (El usuario podrá escribir la aceleración que puede tener cada vehículo)
 - Velocidad máxima (El usuario podrá escribir la velocidad máxima a la que puede moverse el vehículo)

- Color del vehículo (El usuario podrá escoger entre Rojo, Verde y Azul).
- Escribir las funciones, clases y algoritmos para poder realizar la lectura de los archivos descargados por Open Street Maps, y poderlos manejarlos mediante las tres herramientas como son SUMO, Polyconvert y NetConvert.

Todas estas opciones que están listando son fundamentales para poder realizar una simulación altamente apegada a la realidad y es por eso que se deberían incrementar en la versión mejorada de C4R.

2.4 Diagramación UML del Software

En esta sección se definirá los parámetros y pasos lógicos necesarios para el desarrollo del aplicativo, con esto se tendrá una guía para la realización del programa y se podrá cumplir todos los objetivos expuestos en el Capítulo 1.

2.4.1 Fase de Inicio

El objetivo del programa es tener una herramienta más fiable en la realización de simulaciones de tráfico para VANETs, esto es que las simulaciones sean más apegadas a la realidad y puedan intervenir más parámetros de entrada.

2.4.2 Requerimientos Específicos

Para cumplir con los objetivos mencionados en el capítulo y que el software desarrollado en este proyecto de titulación pueda utilizarse de una manera fácil, es necesario considerar los siguientes requerimientos.

- Interfaz de usuario amigable para la elaboración de nuevas simulaciones.

- Incrementar la opción de interacción con los parámetros de POLYCONVERT para realizar la simulación.
- Creación de rutas de manera visual.
- Creación de parámetros de vehículos.
- El software debe ser compatible con cualquier plataforma que tenga Windows 7 o superior

2.4.3 Confiabilidad

El sistema debe ser confiable, para que los resultados que se obtengan de esta simulación sean los más reales y apegados a la realidad, con esto se necesita que los parámetros a considerarse en la entrada y los resultantes a la salida contengan información verídica.

2.4.4 Desempeño

El software debe estar bien dimensionado para que el tiempo de ejecución y respuesta no sea alto y entregue al usuario una agradable utilización del mismo. Para esto se debe tener en cuenta que la programación del mismo debe generar la menor cantidad de variables temporales y evitar los códigos con lazos para no tener pérdidas de memoria y procesador.

2.4.5 Consultas, Informes y entregables

Para poder determinar las clases, funciones y módulos a desarrollarse primero se debe determinar cuáles son los archivos necesarios en SUMO y los programas que se necesitan.

Requerimiento 1

Descarga y generación de archivo de los mapas mediante la herramienta a ocuparse JMapView (extensión .osm.xml).

Requerimiento 2

Generación del archivo de rutas a tomar los vehículos en la simulación (extensión .rou.xml) en el caso que sea simulación aleatoria.

Requerimiento 3

Generación del archivo de rutas a tomar los vehículos en la simulación (extensión rouper.xml) en el caso de que el usuario defina las rutas.

Requerimiento 4

Visualización de nombres de calles y avenidas en la generación de rutas.

2.5 Modelos de Casos de Uso del software

En el desarrollo y diseño de nuevas aplicaciones una parte importante es definir el diagrama de casos de uso del software.

Como la base del programa es C4R, a partir de esto se va a definir los nuevos usos que va tener el software, no se va a tomar en cuenta los usos o funciones anteriormente definidas en la base del programa como son la definición del escenario, creación de directorios, entre otros.

El usuario sin ninguna restricción puede ingresar al sistema y tiene tres opciones para continuar, esto es realizar una nueva simulación, editar una anterior y eliminar una simulación. Tal como se muestra en la Figura 2.1.

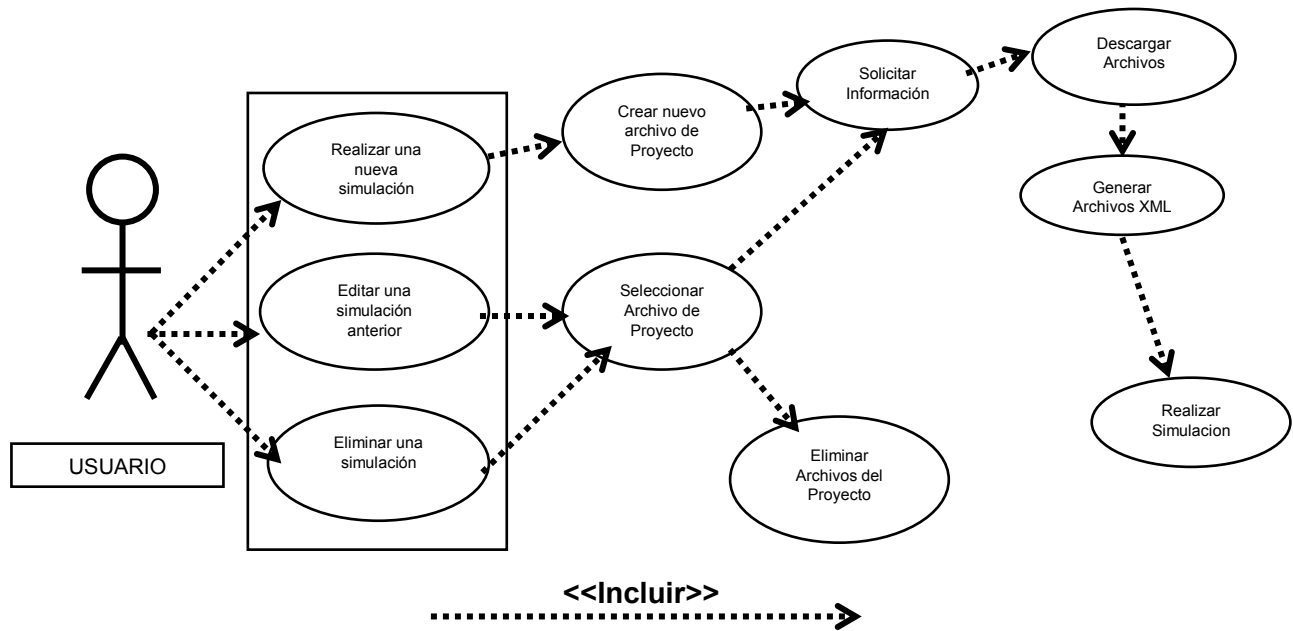


Figura 2.1. Diagrama de casos de Uso General

2.6 Diagrama de casos de uso

En esta etapa del desarrollo del nuevo aplicativo, se va a mencionar los pasos que se necesitarían para cumplir con los objetivos planteados en el capítulo uno, esto es la serie de pasos lógicos y posibles acciones que el usuario va a tener a lo largo del uso del programa. La tabla 2.1 muestra los posibles casos de uso para el programa, con esto se podrá definir la secuencia de interacciones y con esto la programación del aplicativo.

Tabla 2.1 Casos de uso y posibles interacciones.

Nombre: Descarga de Mapas mediante JMapViewr
Actores: Usuario, Servidor Open Street Maps, Hardware
Función: Descargar mediante una herramienta visual los archivos de mapas de un determinado lugar real.
Descripción: Mediante una interfaz gráfica se tendrá que llamar al software JMapViewr y utilizando este software se podrá definir un lugar físico con cuatro puntos cartesianos mediante el cual se descargará toda la información del lugar. Esto es vías, señalización, edificaciones, etc
Referencia: Ninguna
De requerimientos: Requerimiento 1, 2, 3, 4
Actividades de caso: Descarga de mapas

Evento	Respuesta del Sistema
1. Selección de descarga de mapas mediante Open Street Maps	2. Capturar la selección
	3. Abrir el aplicativo JMapView en pantalla completa.
4. Seleccionar mediante las herramientas gráficas, los cuatro puntos del mapa a descargar.	5. Capturar los datos enviados del programa JMapView.
	6. Mostrar la pantalla de POLYCONVERT (generación de XL) para la ejecución del programa.
7. Seleccionar mediante interfaz gráfica los elementos que se quiere que intervengan en POLYCONVERT.	8. Generar el archivo XML (tymap.xml) para la ejecución del comando POLYCONVERT
	9. Ejecutar el comando POLYCONVERT mediante consola con el archivo typemap.xml
	10. Ejecutar el comando NETCONVERT mediante consola para la generación del archivo de Red
	11. Emitir un mensaje que explique al usuario que debe seleccionar simulación aleatoria o personalizada.
12. Seleccionar el tipo de simulación, puede ser aleatoria o personalizada	13. Si es aleatoria, ejecutar la simulación en SUMO-GUI (Interfaz gráfica de SUMO)
	14. Si es personalizada, presentar la pantalla de personalización para generación de archivos.
Eventos alternativos de caso:	
Respuesta 2:	Esperar la selección del usuario, si no se la tiene enviar un mensaje de usuario.
Respuesta 3:	Si existe un error en la carga del archivo,

	mostrar un mensaje informativo de lo que está sucediendo.
Respuesta 5:	Mostrar las limitantes de la descarga de los archivos, mediante mensaje de error. Estos limitantes está en el tamaño mínimo y máximo del área de la ciudad.
Respuesta 6:	Esperar la interacción del usuario para la generación del archivo typemap para la ejecución del programa POLYCONVERT.
Respuesta 8:	Si la generación del archivo typemap fue correcta no notificar al usuario, caso contrario notificar al usuario para su nueva generación.
Respuesta 9:	Si los parámetros necesarios para la ejecución del POLYCONVERT son correctos ejecutar el comando, caso contrario notificar al usuario.
Respuesta 10:	Si los parámetros necesarios para la ejecución del NETCONVERT son correctos ejecutar el comando, caso contrario notificar al usuario.
Respuesta 11:	Tomar el valor de la selección del usuario para la presentación de los cuadros de di
Respuesta 13:	Tomar el valor de la selección de usuario, generar el archivo de configuración y realizar la simulación con los valores por defecto
Respuesta 14:	Mostrar al usuario la ventana de personalización, controlar cada uno de los componentes y botones para evitar errores. Si no se tienen errores realizar la simulación.

2.7 Diagrama de Componentes

Con este tipo de diagramas se puede mostrar las dependencias que tienen los componentes del sistema en tiempo de compilación de enlace en un programa y en tiempo de ejecución entre componentes.

Esto quiere decir que para que el sistema funcione correctamente deben estar presentes todos los componentes sean de compilación o ejecución, estos deben estar funcionando sin errores y siempre estar listos para su uso.

El diagrama de componentes en tiempo de ejecución es el que se muestra en la figura 2.2, también se muestra el diagrama de componentes en tiempo de compilación, este se lo ve en la figura 2.3.

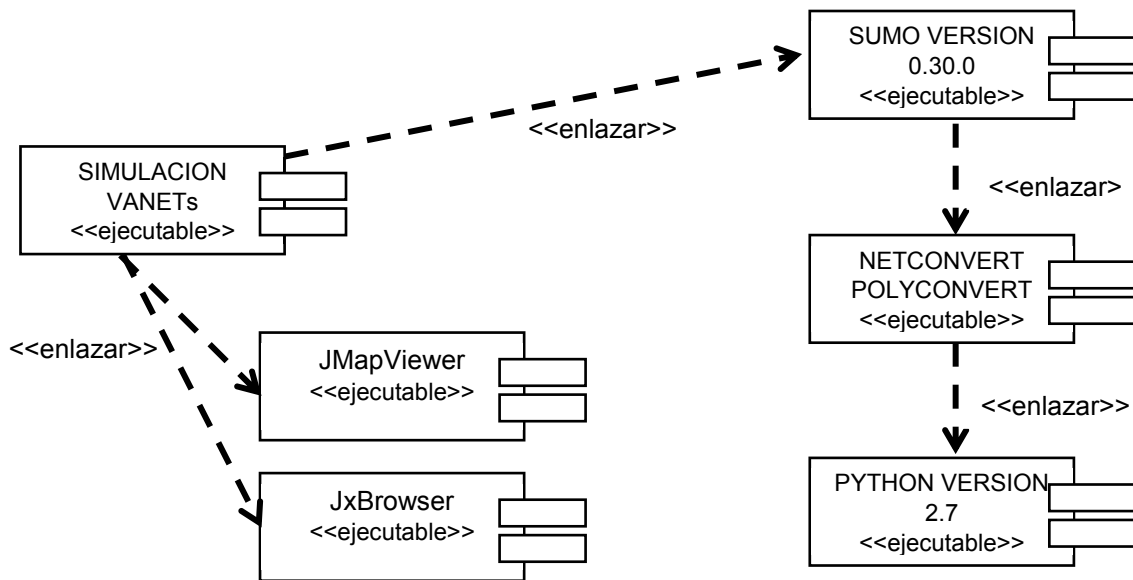


Figura 2.2 Diagrama de componentes que muestra la dependencia en tiempo de ejecución.

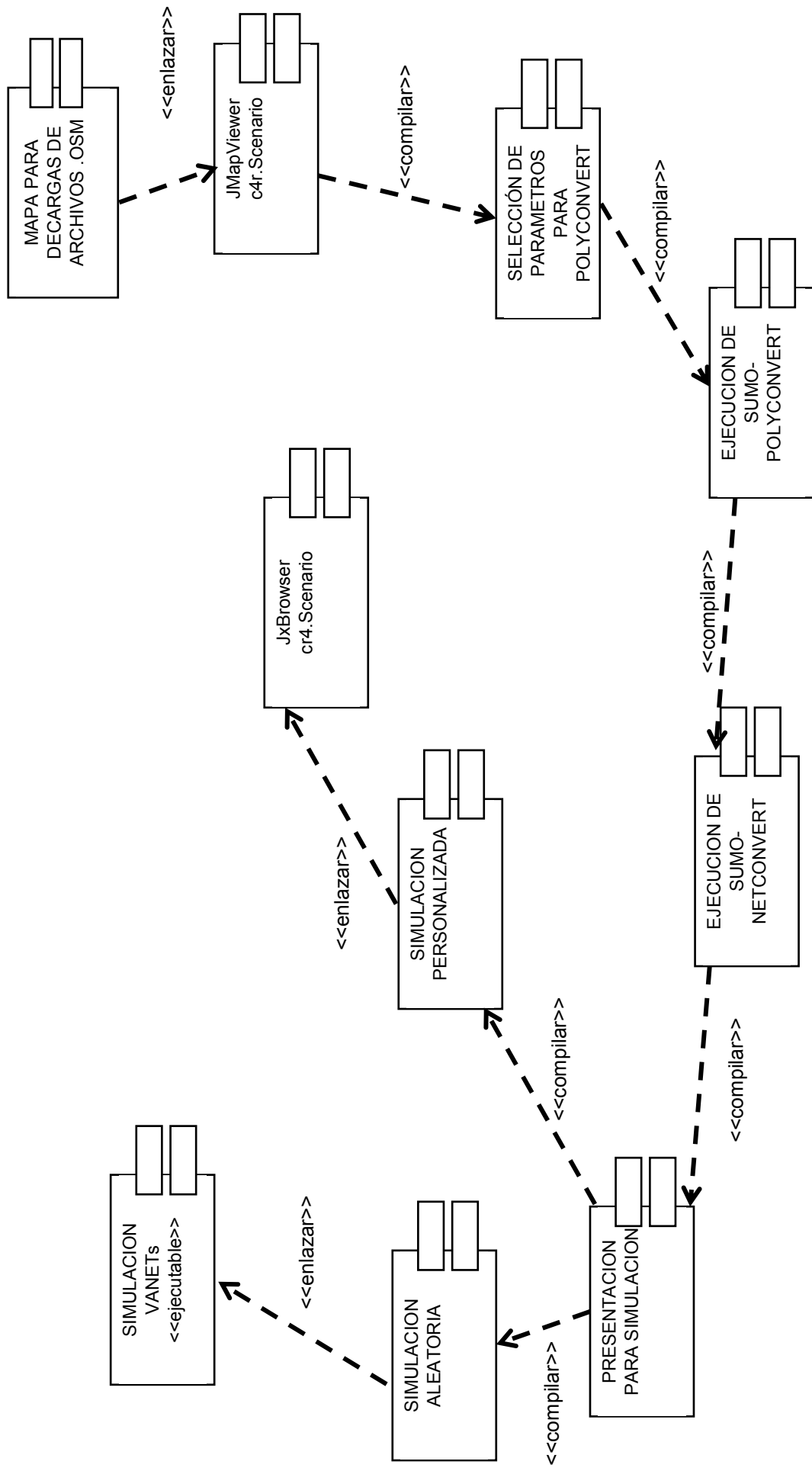


Figura 2.3 Diagrama de componentes que demuestra la dependencia en tiempo de compilación

Del diagrama de la figura 2.2, se puede observar que C4R en su versión mejorada utiliza 4 software que deben estar instalados en el equipo donde va a funcionar la nueva versión de C4R. Estos software son necesarios para cumplir con todos los procesos en la simulación.

- **JMapViewer:** Software para enlazarse con el servidor de Open Street Maps para poder descargar los archivos de mapas.
- **JxBrowser:** Software en java necesario para poder visualizar las vías escogidas en el aplicativo mediante consultas html al servidor de OSM.
- **SUMO:** Software necesario para poder realizar las simulaciones propiamente dichas.
- **NetConvert y Polyconvert:** Software consideradas herramientas de Sumo.
- **Python:** Software necesario para poder ejecutar los scripts de SUMO (Simulaciones Aleatorias).

2.8 Diagrama de navegación

Definido las dependencias y los módulos que se necesitan para la elaboración del software se realizó el diagrama de navegación, que son los posibles caminos que puede seguir el usuario. Este diagrama es importante para la programación por cuanto define las ventanas y pasos de programación, así como las clases a realizarse. La figura 2.4 muestra el diagrama de navegación del software.

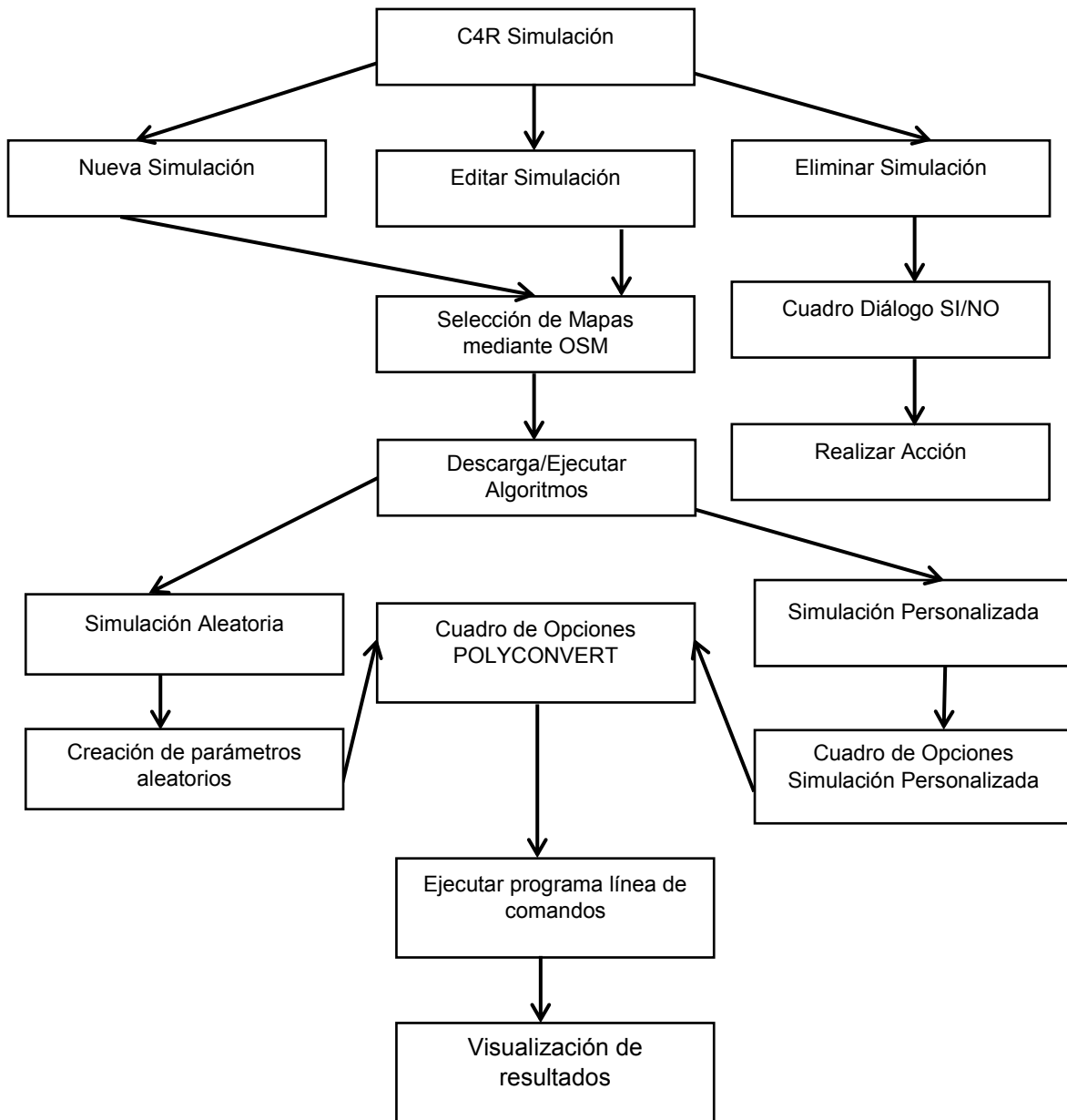


Figura 2.4. Diagrama de navegación

2.9 Diagrama de Clases

Siguiendo con la metodología de Kendall, se va a definir las clases necesarias para la elaboración del software, este diagrama es importante porque en este se van a definir las clases necesarias para la codificación del software a programarse no se consideran las actuales; en el la figura 2.5 se observa el diagrama de clases generado en NetBeans.

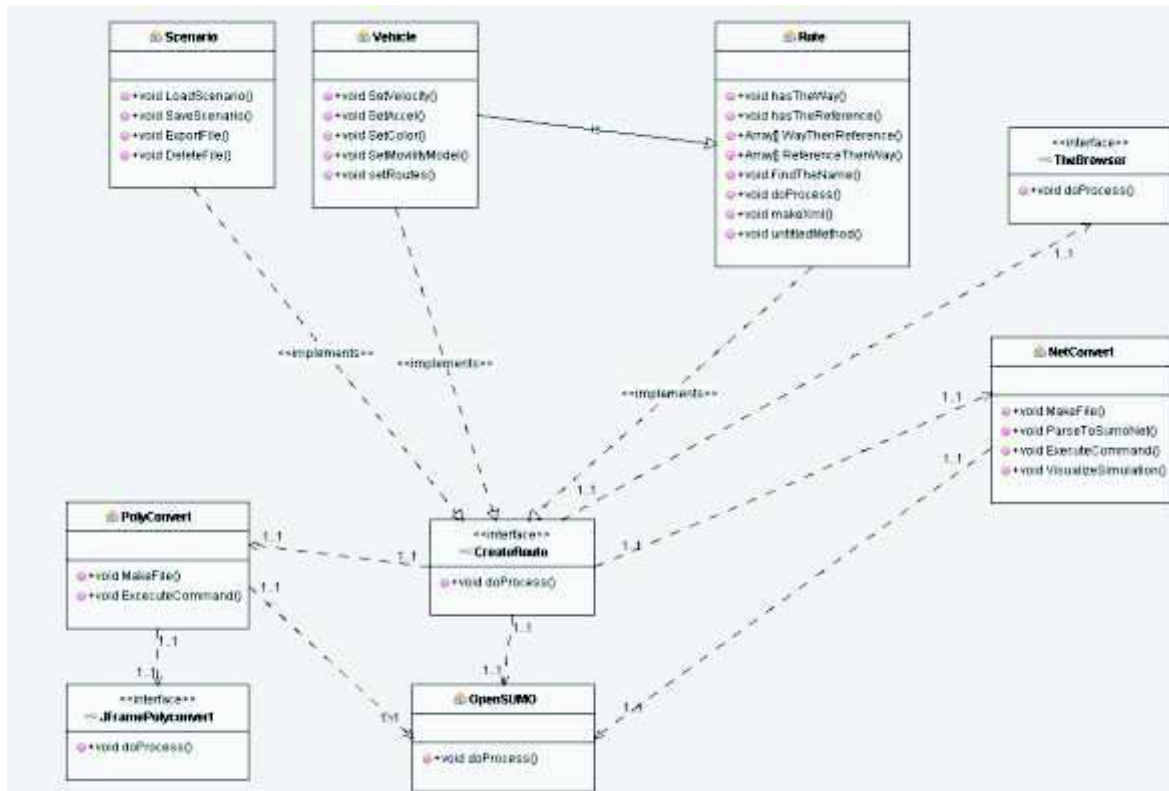


Figura 2.5. Diagrama de Clases.

2.10 Codificación y programación del módulo de simulación

Para cumplir con los requerimientos y alcance de la tesis, se analizó el programa actual de C4R.

De este análisis se pudo distinguir que partes del código se seguirán ocupando, que partes del código van a eliminarse y que se necesita programar desde cero.

Además de lo que se mencionó en la introducción de esta tesis se van a tener dos opciones para mejorar las simulaciones, la primera es la opción de realizar una simulación aleatoria, con parámetros preestablecidos y la segunda es realizar una simulación con parámetros personalizados.

2.10.1 Generación de Simulaciones con parámetros aleatorios

Para hacer este proceso se ocupa el archivo “randomtrips.py” que es una herramienta que se encuentra en la carpeta “tools” de SUMO.

Este archivo randomtrips.py es un archivo escrito en Python, que genera una serie de rutas aleatorias y considera varios parámetros adicionales como atenuaciones, cantidad de tráfico, peatones, vehículos con características aleatorias que se muevan dentro de las vías del archivo OSM descargado.

Para realizar este proceso se necesita ejecutar dos tipos de comandos de Python desde la consola de comandos de Windows.

Adicionalmente cabe mencionar que la versión a ser utilizada de SUMO, puede discriminar las vías que son exclusivamente para vehículos y no utilizaría vías de peatones o vías de tren para enviar tráfico vehicular en las simulaciones.

Estos comandos son los que se muestran en la porción de código de la figura 2.6 de a continuación.

```
python path_of_sumo/tools/trip/randomTrips.py -n map.net.xml -e 100 -l  
python path_of_sumo/tools/trip/randomTrips.py/randomTrips.py -n  
map.net.xml -r map.rou.xml -e 100 -l
```

Figura 2.6 Porción de código para ejecución de randomtrips.py

En un ejemplo real estos códigos deben ejecutarse desde la línea de comandos, de DOS tal como se muestra en la figura 2.7.

En el ejemplo se muestra la ejecución de los archivos de Python desde línea de comandos. Estos archivos se encuentran ubicados en la carpeta:

C:/Users/SantiagoOate/Documents/test/.

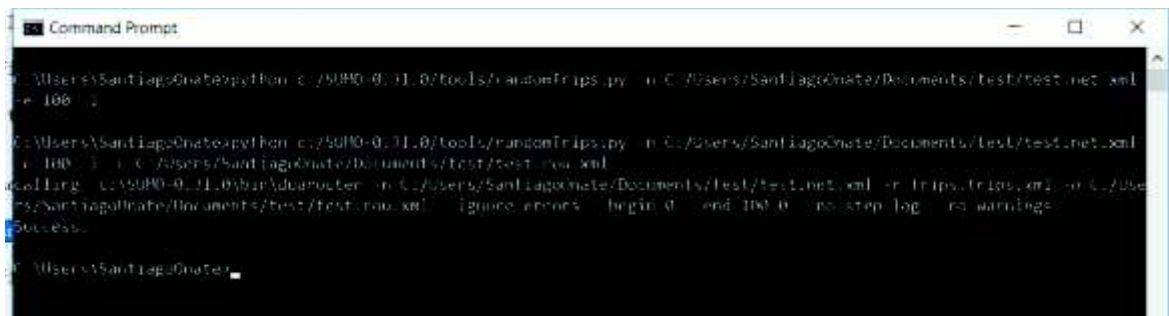


Figura 2.7 Ejecución real de comando de Python

Adicionalmente una de las principales partes del código que también se necesitará modificar es el archivo relacionado a la descarga de los archivos OSM que se obtiene del

software JMapView, por cuanto estos archivos también cambiaron en su composición y estructura.

2.10.2 Generación de Simulaciones con parámetros personalizados

Una opción mejorada del programa y que se encuentra en el alcance de la tesis es poder realizar las simulaciones personalizadas con la mayor cantidad de parámetros de entrada generados por el usuario.

Para esto se dio al usuario la capacidad de escoger y modificar los parámetros de los siguientes puntos:

- Creación y modificación de archivos XML para la ejecución de POLYCONVERT, estos parámetros pueden ser (zonas industriales, zonas naturales, zonas comerciales, etc.).
- Creación de Vehículos con características físicas.
- Asignación de modelos de movilidad vehículo por vehículo, según sea necesario.
- Creación de Rutas.

2.10.3 Manejo de archivos OSM descargados con JMapView

Para poder explicar el trabajo realizado se va a mencionar una parte del archivo generado por JMapView y Open Street Maps.

La siguiente porción de código de la figura 2.8 fue extraída del archivo OSM generado por la herramienta JMapView, en la cual podemos observar en su estructura las siguientes partes:

```
<way id="24214589" visible="true" version="12" changeset="39847043"
timestamp="2016-06-06T19:33:09Z" user="FoxyMarcie" uid="3321091">
  <nd ref="262190260"/>
  <nd ref="262190259"/>
  <nd ref="262188177"/>
  <nd ref="262188176"/>
  <nd ref="262188175"/>
  <nd ref="262188172"/>
  <nd ref="262185781"/>
  <nd ref="262185761"/>
  <nd ref="262185754"/>
```

```

<nd ref="2914625535"/>
<nd ref="4207396819"/>
<nd ref="262185748"/>
<nd ref="262185743"/>
<nd ref="262185717"/>
<tag k="alt_name" v="Oe4"/>
<tag k="highway" v="residential"/>
<tag k="name" v="Conde Ruíz de Castilla"/>
<tag k="source" v="Yahoo! Aerial Imagery"/>
<tag k="surface" v="asphalt"/>
</way>

```

Figura 2.8. Porción de código del archivo OSM

- **way id**

Es el identificativo de cada vía en el archivo de datos, este identificativo servirá más adelante para poder enviar a un vehículo por una vía determinada y extraer las referencias a otras vías.

- **nd ref**

Como su propia etiqueta lo dice, son las referencias de esta vía que tiene hacia otras vías, es decir, mediante este identificativo se puede saber que la vía “AB”, tiene conexión con la vía “BC” y “BD” y no tienen conexión con las vías “DE”, “CF” y “CG”, tal como se muestra en la figura 2.9

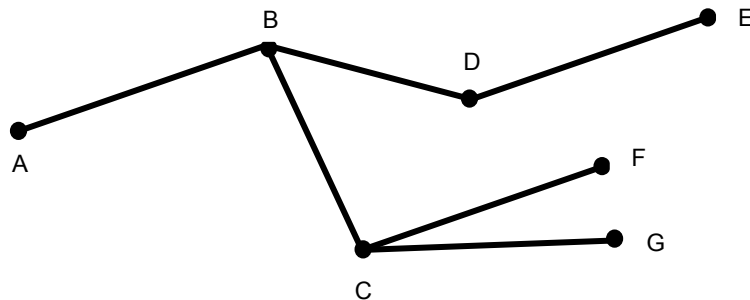


Figura 2.9. Ejemplo de relación de nodos en archivo OSM

Esto es muy importante ya que el usuario podría decirle al vehículo siga por la ruta AB luego vaya por la BD y termine por la ruta DE, pero no puede moverse por la ruta AB luego por la ruta BD y luego por la CG ya que esta última no tiene conexión a la anterior.

Adicional a esto se deberá determinar si las vías que se pueden escoger tienen “conexión” es decir que si es factible trasladarse a una nueva posición, por ejemplo si el

vehículo esta en BC según el gráfico podría saltar a CF o CG, pero si CG es contra vía el vehículo no podrá realizar ese salto y el único posible salto sería CF.

- **tag k= "name" v="Conde Ruiz de Castilla"**: Es el nombre de la vía, que servirá para poder desplegar el nombre en el aplicativo.
- **tag k="highway" v="residential"**: Es el tipo de la vía, con estas opciones se podría discriminar según el tipo de vías de alto tráfico o bajo tráfico, además de esto se podrá determinar la velocidad máxima, esto se podría mejorar en próximas mejoras del proyecto realizado.
- **tag k="alt_name" v="Oe4"**: En ciertas vías se puede encontrar un nombre alternativo que puede también ocuparse según sea el caso.

2.11 Algoritmos y clases para la programación

Como se mencionó en los capítulos anteriores, una desventaja del programa C4R en la versión actual es que utiliza SUMO en la versión 0.11, la versión actual es la 0.30, para poder acoplarse con esta versión se tuvieron que realizar algoritmos de búsqueda en archivos, algoritmos de creación de archivos y algoritmos para ejecución de archivos con parámetros de entrada y salida, para posteriormente ejecutarlos en línea de comandos de *Windows Command Prompt* de DOS.

Un paso importante y previo a la realización de los algoritmos es el análisis del archivo generado por Open Street Maps. Este archivo maneja una estructura XML, en el cual se tiene que trabajar entre los elementos y atributos.

Los elementos que se encuentran al abrir el archivo osm.xml son los que se encuentran etiquetados con los tags "Way" y "Node", los que se necesitan para poder encontrar todas las vías que pueden invertir en la simulación son los Elementos con etiqueta "Way".

Cada uno de estos además tienen unos subelementos que indican una información adicional como tipo de carretera (autopista, calle de primer orden, segundo orden, tipo de terminado de la carretera asfaltada, lastrada) nombre, nombre alternativo, una vía o doble vía entre otras).

Entonces identificados todos los elementos y atributos se realizaron los algoritmos descritos en los siguientes puntos.

2.11.1 Algoritmos de búsqueda en archivos

Como se mencionó una parte importante en la elaboración de la tesis es poder interpretar y analizar los archivos de OSM descargados, para esto se implementó un algoritmo mediante lazos mediante “for” y “do/while”.

Estos lazos fueron creados para que se lean todos los nodos que contengan “way id”, todos estos nodos se vayan almacenando en una lista y a su vez se vayan añadiendo a un JComboBox para que el usuario pueda seleccionarlos de una manera fácil y rápida. La porción de código de la figura 2.10 y 2.11 muestra el proceso mencionado.

```
public void fillcomboboxFirst()
{
    //Clase creada para discriminar si la via tiene o no nombre
    //y ademas si si puede moverse un vehículo por esta
    MyResult info=new MyResult();// ArrayList<String>();
    //Path para grabar los archivos en el directorio de trabajo
    String path=thepath;
    SAXBuilder builder = new SAXBuilder();
    //Leer el archivo descargado de OSM
    File xmlFile = new File(path.substring(0, path.length() - 8)+".osm");
    try {
        Document document = (Document) builder.build(xmlFile);
        Element rootNode = document.getRootElement();
        //Hacer una lista con todos los elementos que tienen way
        List list = rootNode.getChildren("way");
        for (int i = 0; i < list.size(); i++) {
            Element node = (Element) list.get(i);
            //Hacer una lista con todos los elementos que tienen tag
            List listtag = node.getChildren("tag");
            //numero de elementos que tienen tag
            int sizeoflist=listtag.size()-1;
            int theelement=i;
            info=information(sizeoflist,listtag,theelement);
            int valor;//tiene o no tiene nombre
            int pos;//posicion de etiqueta nombre
            int notway;//puede ser ocupada por vehículo
            valor=info.retunrhavename();
            pos=info.retunrpos();
            notway=info.retunrnotway();
            if(pos<=2)
                pos=2;
            Element tag = (Element) listtag.get(pos-2);
            if(valor==1)//Si tiene nombre
            {
                String nameofstreet;

                nameofstreet=tag.getAttribute("v").toString().substring(15);
                nameofstreet=nameofstreet.substring(0,
                nameofstreet.length() - 2);
                String idofstreet;

                idofstreet=node.getAttribute("id").toString().substring(16);
                idofstreet=idofstreet.substring(0, idofstreet.length() -
                2);

                listOsmWays.add(nameofstreet+"/"+idofstreet);
            }
        }
    }
}
```

```

        cmbRoutes.addItem(nameofstreet+"/"+idofstreet);//+ "-
"+lareferencia);
    }
    else if(valor==0&&notway==0)//No tiene nombre pero si es para
simulacion
    {
        String idofstreet;
idofstreet=node.getAttribute("id").toString().substring(16);
        idofstreet=idofstreet.substring(0, idofstreet.length() -
2);
        cmbRoutes.addItem("No information"+"/"+idofstreet);// + "-
"+lareferencia);
        listOsmWays.add("No information"+"/"+idofstreet);
    }
    }
}
catch (IOException io) {
    System.out.println(io.getMessage());
} catch (JDOMException jdomex) {
    System.out.println(jdomex.getMessage());
}
}

```

Figura 2.10 Porción de código de algoritmo para leer archivos XML de OSM

```

public MyResult information(int value, List listtag, int theelement)
{
    //Inicialización de variables con valores aleatorios
    int havename=-1;
    int position=-1;
    int notway=-1;
    MyResult theresult=new MyResult();
    for(int i=0;i<listtag.size();i++)
    {
        String nodotag;
        Element tag = (Element) listtag.get(value);
        nodotag=tag.getAttribute("k").toString();
        if(nodotag.contains("name"))//si tiene el el nombre de la vía
        {
            havename=1;//Si tiene nombre de la vía
            position=i;//Posicion del elemento tag que tiene el nombre
            notway=0;//si es una vía para vehículo
        }
        //Discriminar que no se utlice las vías y edificios en las
        //simulaciones
        else if(nodotag.contains("building")
            ||nodotag.contains("train")
            ||nodotag.contains("train")
            ||nodotag.contains("pedestrian")
            ||nodotag.contains("path")
            ||nodotag.contains("bridleway")
            ||nodotag.contains("cycleway")
            ||nodotag.contains("footway"))
        {
            havename=0;//no tiene nombre de vía
            position=i;//posición
            notway=1;//no es una vía para simulación
        }
        else
        {
            havename=0;//No tiene nombre de la vía

```

```

        position=i;//Posición
        notway=0;//Si es una vía para simulación
    }
    value--;
}
//Enviar los resultados para ser ocupados
theresult.havename=havename;
theresult.position=position;
theresult.notway=notway;
//Devolver la variable para escritura de resultados
return theresult;
}

```

Figura 2.11 Porción de código para escribir nombre de vía y discriminar tipo de vía

Una parte importante del código de la figura 2.11 entre las líneas 21 a 28, es que este discrimina las vías exclusivas para peatones, para trenes, para vehículos y con esto se cumplirá con que los parámetros de las simulaciones son los mas reales posibles.

La finalidad de este algoritmo es cargar en memoria una lista con todas las vías que se dispone en el archivo OSM descargadas mediante jMapViewer y además llenar el JComboBox.

La estructura resultante del String formado como elemento del JComboBox es el siguiente:

- a) Si la vía tiene nombre:
Nombre de la vía + "/" + idVia
- b) Si la vía no tiene nombre:
"No information" + "/" + idVia

2.11.2 Algoritmos de creación de archivos

También como una mejora en la simulación y por ende que esta sea más real se realizó una sección de programación mediante la cual el usuario de una manera gráfica va a crear las rutas que van a seguir un determinado número de vehículos.

Esto se divide en dos partes:

- La primera es crear el vehículo en sí relacionado a las características físicas.
- La segunda es poder crear una ruta específica.

Como se mencionó, para realizar este proceso lo que se realizó es extraer todas las vías que se encuentran en el archivo generado por JMapView mediante el algoritmo mencionado anteriormente.

Al estar añadidas todas las vías del archivo OSM en el JComboBox, mediante el algoritmo de la figura 2.12 se pudo discriminar que vías están conectadas con la primera vía seleccionada y con esto solo se podrán formar rutas reales que puedan seguir los vehículos. Si las vías no pueden conectarse directamente el programa trata de llenar los saltos intermedios que se debe tener.

```
private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    //Creación de una array de string para enviar todas las referencias
    //que tiene una determinada vía (la seleccionada en el JComboBox)
    ArrayList<String> thereference=new ArrayList<String>();
    String route;
    String way;
    //Extraer el string seleccionado en el JComboBox
    route=cmbRoutes.getSelectedItem().toString();
    //Solo el string del id de la vía
    way=route.substring(route.lastIndexOf("/") + 1);
    //Se llama a la función que devuelve la vía dada la referencia
    thereference=waythenreferences(way,0);
    //Borramos todo el contenido del combobox
    cmbRoutes.removeAll();
    //Funcion para llenar nuevamente el combobox y la lista de rutas
    //creadas
    doprocess(way);
    //Llena la lista gráfica para la creación de la ruta
    lstRoutes.add(way);
}

public ArrayList<String> waythenreferences(String theway, int pos) {
    //El path donde se encuentra todo el trabajo
    String path=thepath;
    //Código para leer archivos xml, e interpretarlos
    SAXBuilder builder = new SAXBuilder();
    File xmlFile = new File(path.substring(0, path.length() - 8)+".osm");
    //Creación de listas para leer los datos de las vías
    ArrayList<String> thereference=new ArrayList<String>();
    ArrayList<String> justthereference=new ArrayList<String>();
    //Lectura de los archivos xml, y con opción de "catch" errores
    try {
        Document document = (Document) builder.build(xmlFile);
        Element rootNode = document.getRootElement();
        //Se crea una lista de todas las vías
        List childWay = rootNode.getChildren("way");
        for(int j=0;j<childWay.size();j++)
        {
            Element node = (Element) childWay.get(j);
            //Se crea una lista de todas las referencias de cada vía
            List ndReference = node.getChildren("nd");
            if(node.getAttribute("id").toString().contains(theway))
            {
                for(int item=0;item<ndReference.size();item++)
                {

```



```

        writer2.println("</routes>");

    } catch (FileNotFoundException ex) {
        Logger.getLogger(JFrameRoutes.class.getName()).log(Level.SEVERE,
null, ex);

        } catch (UnsupportedEncodingException ex) {
        Logger.getLogger(JFrameRoutes.class.getName()).log(Level.SEVERE,
null, ex);
    } finally {
        writer2.close();
    }
}
}

```

Figura 2.13 Porción de código que genera el archivo de rutas personalizado

2.11.3 Algoritmos de ejecución de archivos

Teniendo ya creados los archivos, estos serán ejecutados en el *Command Prompt* de Windows de la siguiente manera.

- Se escribirá un *String* con todos los parámetros necesarios para la ejecución de la orden.
- Se validará que la estructura y escritura del mismo sea correcto.
- Se ejecutará en el *Command Prompt* de Windows mediante funciones específicas de Java.
- Se mostrarán mensajes de ejecución incorrecta si fuese el caso, si se ejecutó de manera correcta se mostrarán los resultados o visualización obtenida.

Un ejemplo de este proceso se incluye en la figura 2.14 y se muestra a continuación.

```

public boolean parseToSUMONet() throws IOException, InterruptedException,
SUMOException {

    String path=thepath;
    ///SUMO netconvert v0.31.0
    final String cmd = " --osm-files " + path + " -o " + pathNetXML;
    String execution = "";
    if (SOValidator.isWindows()) {
        AppConfiguraton conf = new AppConfiguraton();
        conf.load();
        execution = conf.getSumoPath() + "\\netconvert.exe" + cmd;
    } else if (SOValidator.isUnix()) {
        execution ="netconvert" + cmd;
    }
    System.out.println ("Esto se ejecuta:::::" + execution);
    Process p = Runtime.getRuntime().exec(execution);

    /*Cleans the output and error buffers to avoid hang the process.*/

```

```

BufferCleaner bcOut = new BufferCleaner(p.getInputStream());
BufferCleaner bcError = new BufferCleaner(p.getErrorStream());

bcOut.start();
bcError.start();
int outValue = p.waitFor();

if (outValue != 0) {
    /*Error occurred*/
    String nameLog = path.substring(0, path.length() - 8) +
Extensions.LOG.getExtension();
    PlainTextIO.save(nameLog, bcError.getOutput());
    throw new SUMOException("Impossible convert the map to SUMO
format.\n"
        + "Further information in " + nameLog);
}
return outValue == 0;
}

```

Figura 2.14 Ejecución de NETCONVERT en el *Command Prompt* de Windows

Con estos procesos, se generarán todos los archivos y datos necesarios para poder ejecutar el SUMO-GUI y poder lanzar la visualización de los datos en el entorno gráfico.

Si todos los archivos generados fueron correctamente inicializados se podrá ver la simulación en el entorno gráfico.

2.12 Requisitos de Software y Hardware

2.12.1 Requisitos de Software

Como ya se mencionó en capítulos anteriores, para el correcto funcionamiento de C4R se necesita que el software SUMO 0.30.0 y Python 2.7 estén instalados en el directorio raíz del disco duro. Se recomienda instalar como se escribe a continuación.

Directorio de instalador de SUMO: C:\sumo-0.31.0\

Directorio de instalación de Python: C:\Python27\

2.12.2 Requisitos de Hardware

Igualmente, el hardware debe cumplir con mínimos requisitos para un correcto funcionamiento del software a programarse.

Para encontrar los requisitos de hardware lo que se realizo es considerar las características mínimas del Compilador y ver que tanto de recursos del sistema el software programado necesita en el administrador de tareas.

2.12.3 Consideraciones para Memoria y Procesador

Se determinó de la siguiente manera, se encontró cuáles son las características mínimas que se necesita para que un software pueda ser compilado por la versión de *NetBeans* IDE 8.2 de 64bits esto es lo que se observa en la tabla 2.2

Tabla 2.2 Características Mínimas equipo de computación para Netbeans

PC con procesador Pentium Dual Core velocidad de 3200 MHz (o superior).
1024 MB de RAM.
1 GB de espacio de disco duro.
Monitor VGA.

Consideraciones de Aplicativo desarrollado

Adicionalmente considerando pruebas de ejecución del programa se pudo comprobar que no necesita gran cantidad de recursos informáticos como son memoria y procesador, ya que el administrador de tareas revela los siguientes datos.

Memoria Usada: 54.3 MB RAM

% de CPU utilizado: 0.1% de un CPU Intel Core i5 3.0Ghz

Consideraciones de aplicaciones requeridas

Como se mencionó anteriormente C4R necesita del software SUMO-0.30.0 y del software Python. Igualmente se miden los recursos informáticos de memoria y procesador de cada uno de estos y se determina que:

Python

Memoria Usada: 25.2 MB RAM

% de CPU utilizado: 0.1% de un CPU Intel Core i5 3.0Ghz

SUMO 0.30.0

Memoria Usada: 40.0 MB RAM

% de CPU utilizado: 4.0% de un CPU Intel Core i5 3.0Ghz

Consideraciones de espacio en disco duro

Por último, para determinar el tamaño necesario para almacenar el aplicativo se consideran los tamaños de los tres componentes que intervienen en el software desarrollado, el tamaño del programa desarrollado, el tamaño del Software Python y el tamaño del software SUMO 0.30.0. De esto se tiene que se necesita un espacio de alrededor de 620MB.

Entonces tomando en cuenta todas las consideraciones anteriormente expuestas se llega a la conclusión que los requerimientos mínimos de Hardware son las que se muestran en la tabla 2.3.

Tabla 2.3 Características mínimas equipo para simulación

Computadora Con procesador Intel Dual Core de 1.0Ghz
Memoria RAM de 256MB
Espacio mínimo de disco duro 1024MB
Tarjeta de red LAN
Tarjeta Wireless (opcional)
Monitor VGA.

2.13 Manual de Usuario del programa C4R Mejorado

Para poder realizar una simulación lo primero que se tiene que realizar es abrir la aplicación y seleccionar nuevo proyecto tal como se muestra en la figura 2.15



Figura 2.15 Menu principal para realizar una simulación

Después de esto se tendrá que grabar el nuevo proyecto dentro de la ubicación que se decida, esto se lo puede observar en la figura 2.16 tal como se muestra a continuación.

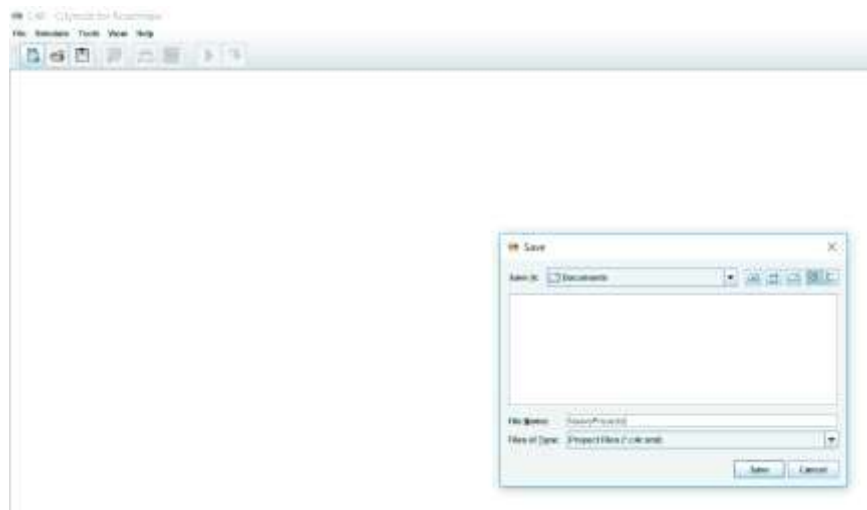


Figura 2.16 Grabar un nuevo proyecto dentro de una carpeta específica

Al guardar el proyecto se habilitará la opción de Seleccionar mapa, en donde se deberá dar clic para poder abrir el buscador de Mapas tal como se muestra en la figura 2.17.

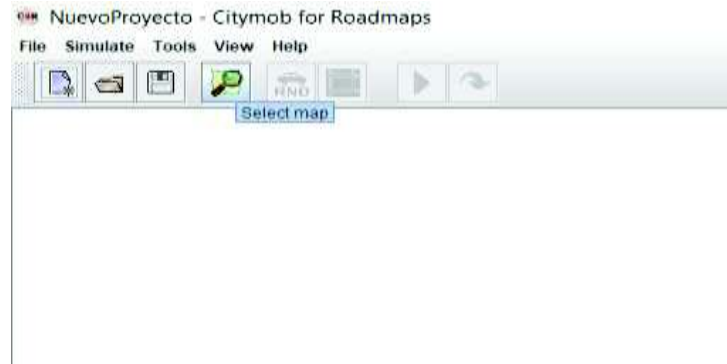


Figura 2.17 Habilitación de la opción Seleccionar Mapa.

Al dar clic en el botón de Seleccionar Mapa, se abrirá el Programa JMapView mediante el cual podrá navegar en un mapamundi, y podrá seleccionar un pedazo de mapa que importará una serie de datos para la simulación. Esto se puede observar en la figura 2.18 que se tiene a continuación.

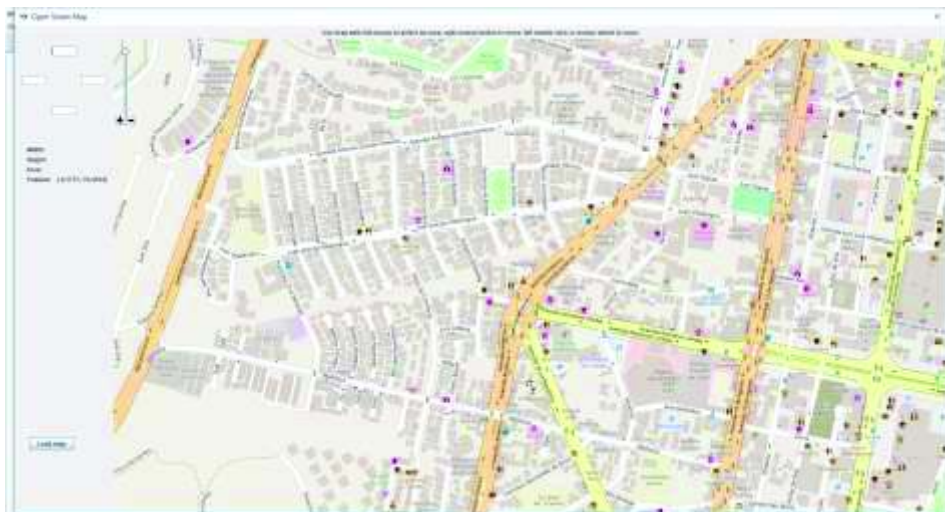


Figura 2.18 Selección de mapa para realizar la simulación

Con el mouse y haciendo doble clic puede acercarse y moverse dentro del mapa para seleccionar el área requerida, se recomienda abrir un área de alrededor de 0.5 km de ancho por 0.5 km de largo. Luego de seleccionar la porción de mapa se pondrá el botón Cargar Mapa, el cual cargará el mapa y grabará un archivo OSM en el directorio principal del proyecto, esto se puede ver en la figura 2.19.

Al descargar el archivo OSM, y si no se presenta ningún error se podrá acceder a la pantalla de generación del archivo de *typemap* para ser ocupado mediante el programa POLYCONVERT, esta opción se la puede observar en la figura 2.20

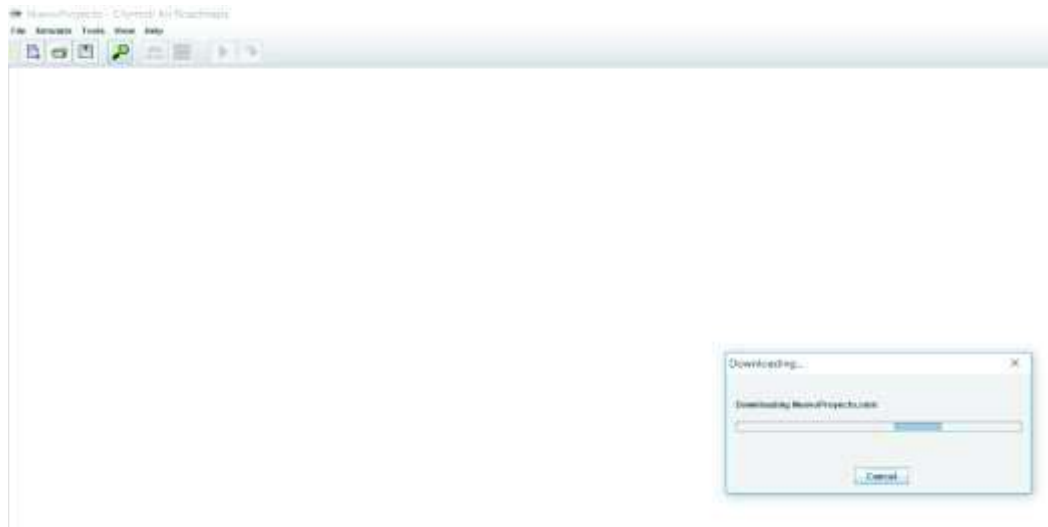


Figura 2.19 Descarga del archivo OSM



Figura 2.20 Opciones de POLYCONVERT para la generación del archivo typemap

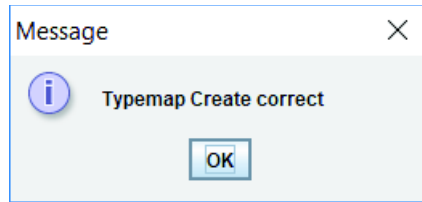


Figura 2.21 Mensaje de creación correcta de archivo typemap.xml

Para realizar una simulación de manera aleatoria en todos los parámetros se tendrá que seleccionar el botón con hacer una Simulación Aleatoria, tal como se muestra en la figura 2.22



Figura 2.22 Selección de Simulación Aleatoria

Al seleccionar la opción de simulación aleatoria solo se tiene que llenar el número de vehículos que se quiere que se simulen. El *valor de la atenuación* y el *depearture* en (s), al poner aceptar se tendrán todos los vehículos que van a intervenir en la simulación. Y se generara el archivo .net y .c4r que serán llamados desde el *command Prompt* de Windows. Esto se puede observar en la Figura 2.23 que se encuentra a continuación.

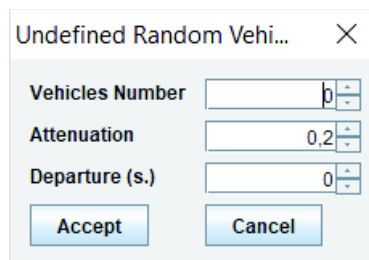


Figura 2.23 Definición de número de vehículos para una nueva simulación aleatoria

Para finalizar y poder comenzar con la simulación se tendrá que escoger la opción Visualizar Simulación Aleatoria, tal como se muestra en la Figura 2.24



Figura 2.24 Selección de Visualización Aleatoria

Al escoger la opción de simulación aleatoria, y si los archivos se generaron sin errores se presentará la pantalla de la Figura 2.25, en donde al aceptar se podrán ya visualizar las simulaciones.



Figura 2.25 Ejecución correcta de parámetros de la simulación

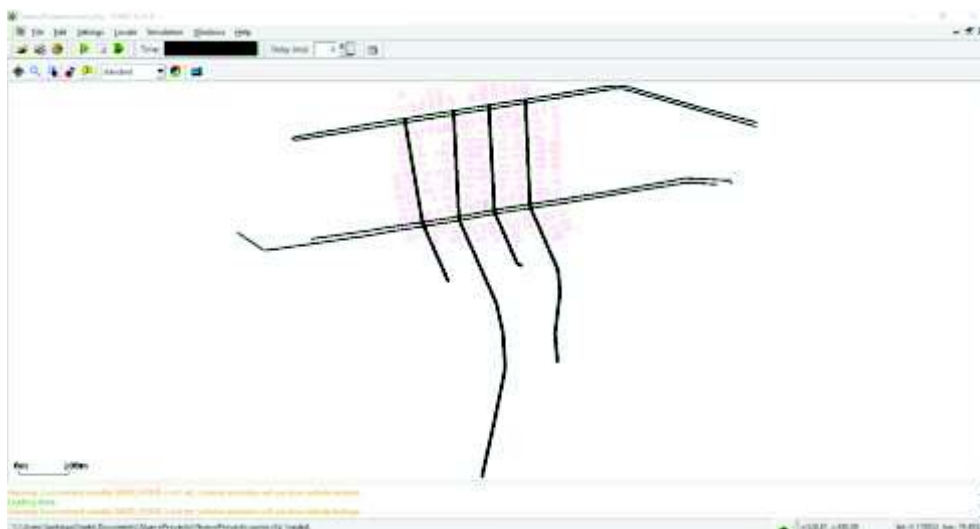


Figura 2.26 Ejecución correcta de parámetros de la simulación

Por ultimo para visualizar el movimiento de los vehículos se tendrá que poner el botón de Play o Ejecutar y podrá observar cómo se mueven los vehículos a lo largo del sistema de rutas.

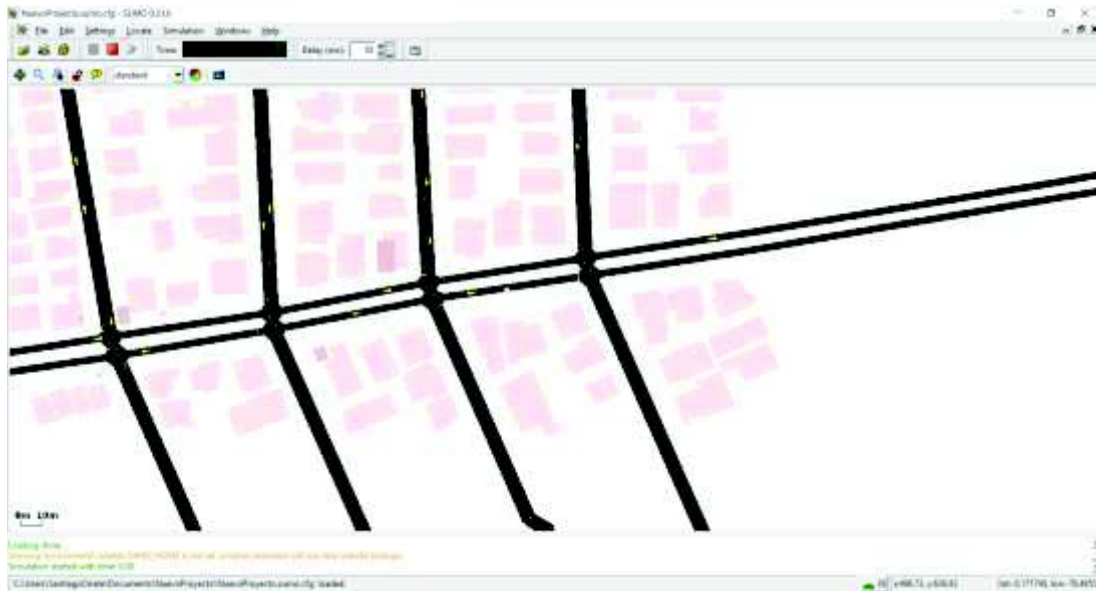


Figura 2.27 Zoom y ejecución de la simulación de manera aleatoria

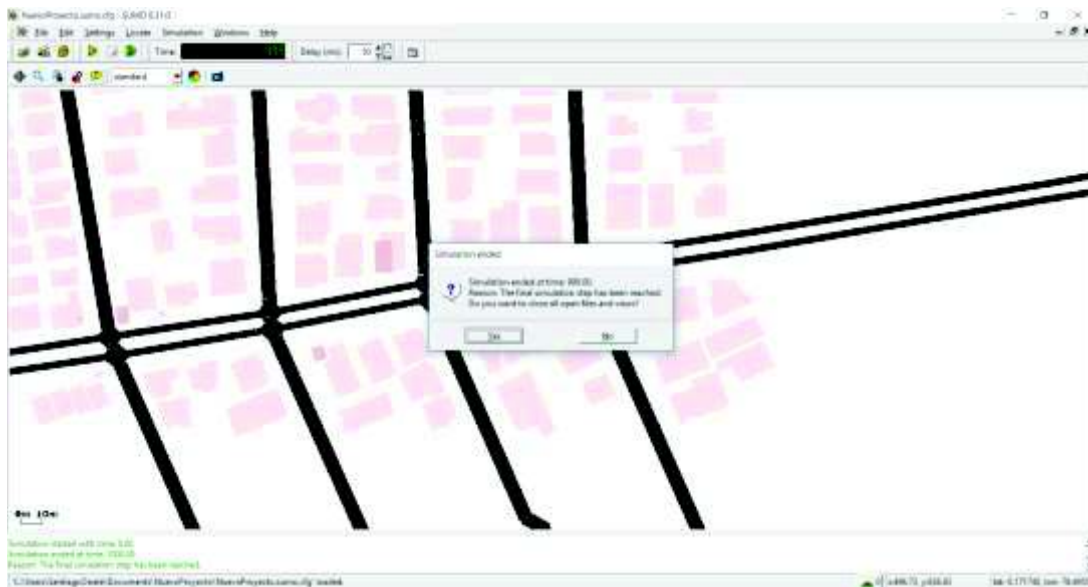


Figura 2.28 Mensaje de advertencia que la simulación termino correctamente

Para realizar una simulación con los parámetros introducidos manualmente se tendrá que seleccionar el botón Personalizar Todo, tal como se muestra en la Figura 2.29.

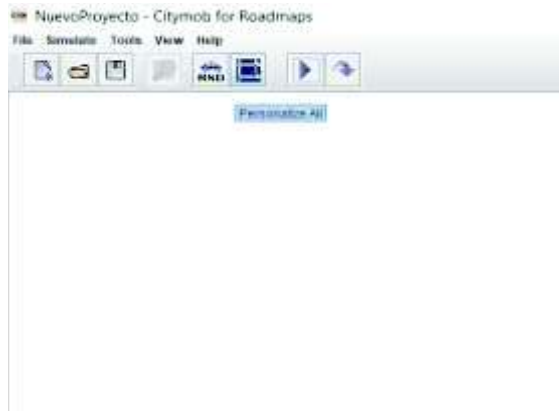


Figura 2.29 Selección de simulación con parámetros personalizados

Al seleccionar esta opción se abrirá la pantalla para poder personalizar los parámetros de los vehículos y las rutas que pueden tomar los vehículos, esta pantalla se muestra en la figura 2.30.

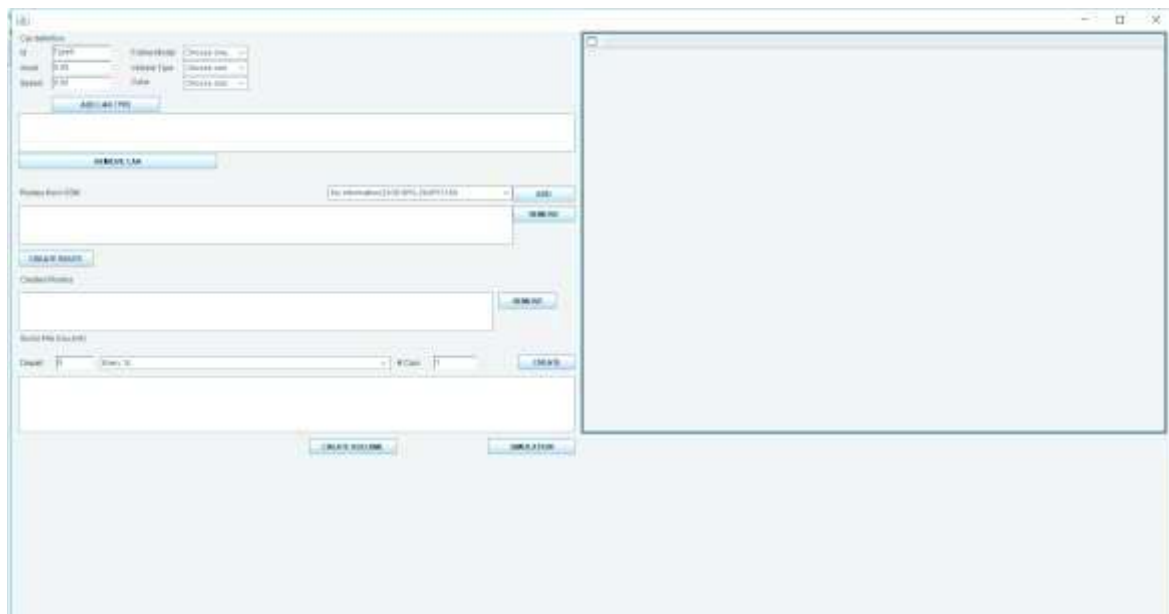


Figura 2.30 Cuadro de diálogo para la elaboración de parámetros personalizados

En la figura 2.30 se pueden distinguir tres componentes, el primero el de la definición del Vehículo, donde de manera gráfica se podrá crear vehículos darles un nombre "Type", una aceleración "accel", modelo de movilidad seguido por el vehículo "Car Following Model", velocidad "Speed" y color.

Al llenar todos estos datos y dar clic en el botón de Añadir Tipo Vehículo se llenará una lista del tipo de vehículo generado, cabe recalcar que se puede crear cualquier cantidad de tipos de vehículos.

También se tiene la opción de remover un vehículo con el botón respectivo. Por ejemplo en la Figura 2.31 se creó un vehículo con nombre “Type1”, que sigue el modelo de movilidad “Krauss”, tiene una aceleración de “5m/s²”, con una velocidad de “100km/h” de color rojo y con tipo “Camión”.

The screenshot shows a 'Car definition' form with the following fields and values:

Id	Type1	Follow Model	Krauss
Accel	5	Vehicle Type	truck
Speed	100.00	Color	red

Buttons: ADD CAR TYPE, REMOVE CAR

Generated code: `vType id="Type0" accel="5" speed="100.00" carFollowModel="Krauss" vClass="truck" color="red"`

Figura 2.31 Creación de un nuevo vehículo

De la Figura 2.30 también se puede observar que en el JComboBox se cargaron las vías que pueden ser ocupadas por el vehículo y se pueden ir llenando y creando una ruta esto es seleccionando del JComboBox y dándole un clic en el botón Añadir, hay q repetir este proceso con todas las rutas que se necesita que se mueva el vehículo, al completar las rutas se da clic en el botón Crear Ruta y se llenara en la lista de Rutas Creadas tal como se muestra en la Figura 2.32.

Cabe recalcar que en la parte derecha se puede ir visualizando la ruta seleccionada en un mapa tal como se muestra en la figura 2.33

The screenshot shows the 'Routes from OSM' form with the following elements:

- Search field: No information/24381850-264553182
- Buttons: ADD, REMOVE
- Route ID: 24381850
- Button: CREATE ROUTE
- Created Routes section:
- Generated code: `route id="route0" edges="-- 24381850 24381850 24381850 --"`
- Button: REMOVE

Figura 2.32 Creación de una nueva ruta

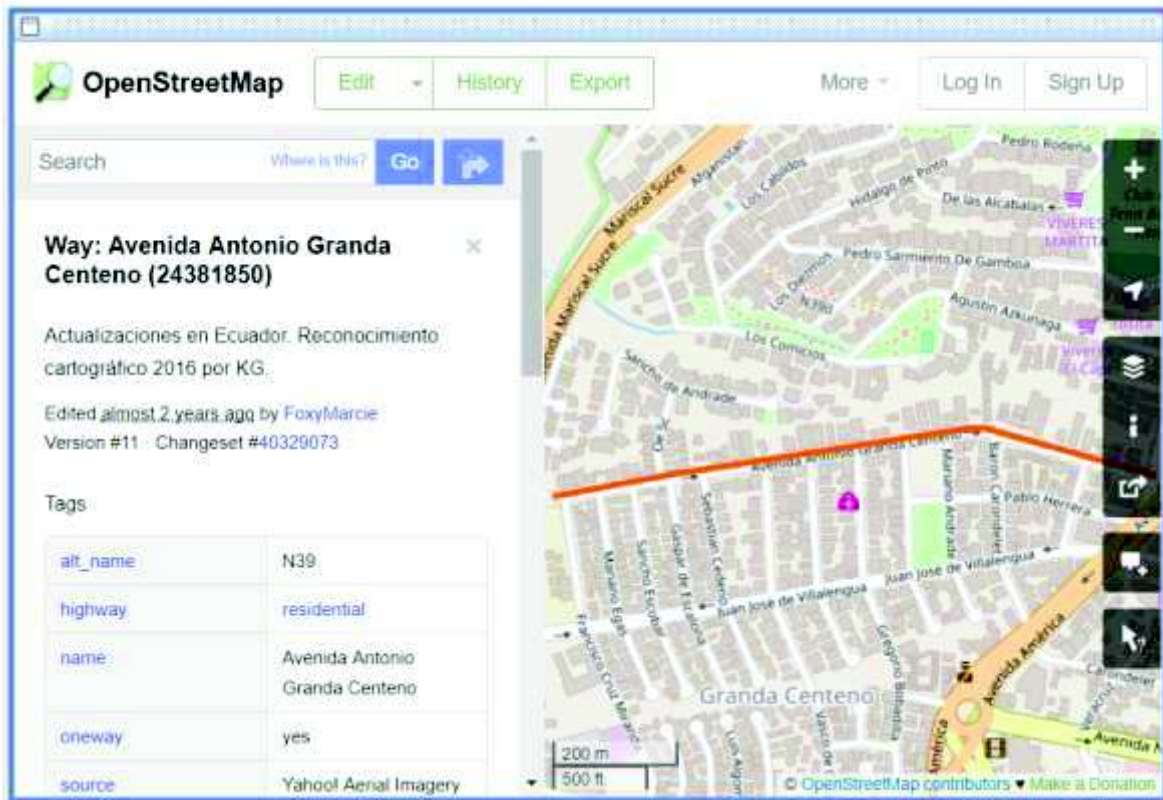


Figura 2.33 Pre visualización de la ruta seleccionada para la generación de rutas

Para la creación del archivo .roupre.xml se necesitara seleccionar un tipo de vehículo, una ruta específica, escribir cuantos vehículos se tienen que crear y cada cuanto van a salir del lugar de origen. Esto se puede ver en la figura 2.34.

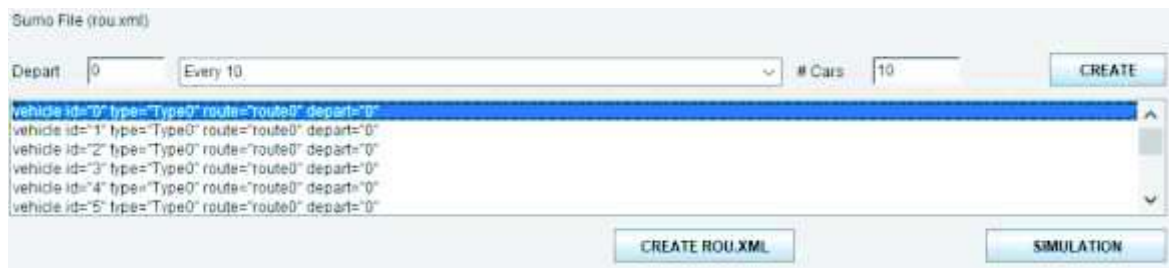


Figura 2.34 Generación de archivo de rutas

Para escribir el archivo de rutas se tendrá que dar un clic en el botón Crear rou.xml y si no ocurre algún error este archivo se podrá utilizar para realizar la simulación que arrancará al presionar sobre el botón Simulación.

Si todo es correcto se crearan los archivos .rouper.xml y c4r.xml para poder generar la simulación personalizada, igualmente se abrirá el ejecutable tal como se mostró en las Figuras 2.27 y 2.28.

3. RESULTADOS Y DISCUSIÓN

En esta sección se evaluará el software desarrollado para las simulaciones de tráfico, este proceso será documentado en todas las secciones para poder encontrar las ventajas y diferencias que se encuentran para llegar a la conclusión que con estas mejoras se tendrán simulaciones más realistas, cabe recalcar que estas simulaciones deberán funcionar en conjunto con simuladores de redes vehiculares VANETs como ns2.

3.1. Resultados

En este capítulo se muestra los resultados obtenidos en simulaciones realizadas mediante el software programado, y el software original, estos resultados se exponen en tablas e imágenes para su análisis y comparación.

Para comenzar con la recolección de información se realizaron simulaciones entre las dos opciones de software, las primeras pruebas se realizaron con la versión mejorada del software C4R para luego hacer las mismas pruebas en el mismo sitio (ubicación en el mapa) con el software C4R en su versión original.

Para realizar las pruebas se tomó una locación en donde se encuentre vías de ferrocarriles, vías de tren o vías exclusivas para transporte pesado que se sabe que en la realidad no pueden ser ocupadas por vehículos de transporte normal.

Para esto se seleccionó un sitio en New York como se muestra a continuación en la Figura 3.1, esto se realizó con total similitud en los dos programas.

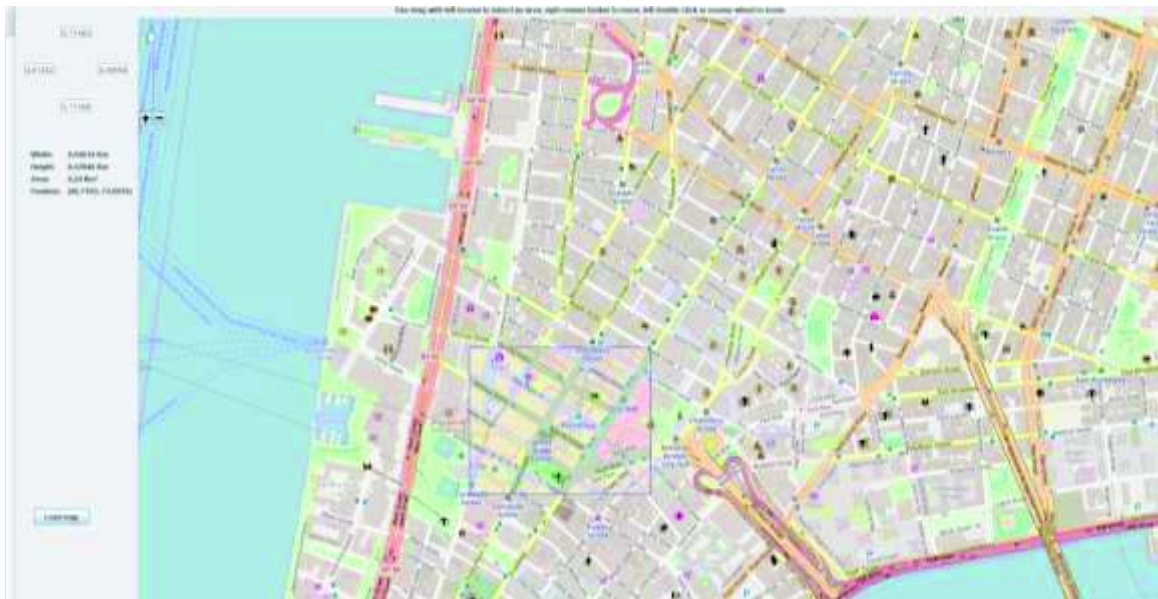


Figura 3.1 Selección de porción de mapa para realizar simulaciones

Versión Mejorada de C4R

Al seleccionar la porción de mapa deseado, el software procederá a descargar toda la información correspondiente a los mapas y archivos necesarios para la simulación. Esto se puede observar en la figura 3.2.

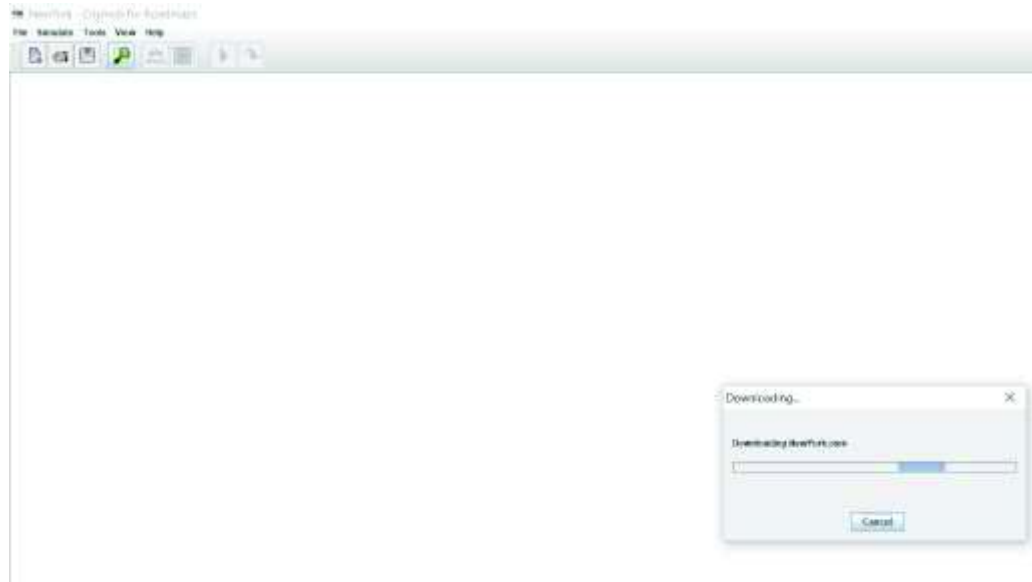


Figura 3.2 Descarga de archivos OSM (Open Street Maps)

Tal como consta en el manual de usuario y como se mencionó anteriormente se va a realizar una simulación aleatoria en todos sus parámetros, entonces se seleccionó dicha opción para comenzar la simulación.

Para el uso de PolyConvert se deberá seleccionar los parámetros que se quiere considerar. Para la prueba correspondiente no se va a seleccionar las siguientes opciones que no son de mucha importancia.

- WaterWay
- Natural
- Natural Water
- Natural Wood
- Natural wetland
- Landuse park

- Landuse forest
- Landuse village_green

Este proceso realizado se lo puede observar en la Figura 3.3 que se muestra a continuación; al presionar el botón ACCEPT se creará el archivo TypeMap.xml y se podrá ejecutar el comando PolyConvert sin ningún inconveniente.



Figura 3.3 Parametros de TypeMap para PolyConvert

Como se han escrito y generado todos los archivos necesarios para la simulación y ya que todo está correctamente configurado, se desplegará el mensaje de la figura 3.4 y posteriormente se abrirá la pantalla de simulación.

Es en este momento donde se puede ver como los vehículos físicamente se van moviendo en cada una de las vías que se encuentran en la simulación.

Para comprobar que la simulación está funcionando correctamente, se seleccionó a la azar dos vías, cuyas características e identificadores se muestran en las tablas 3.1 y 3.2

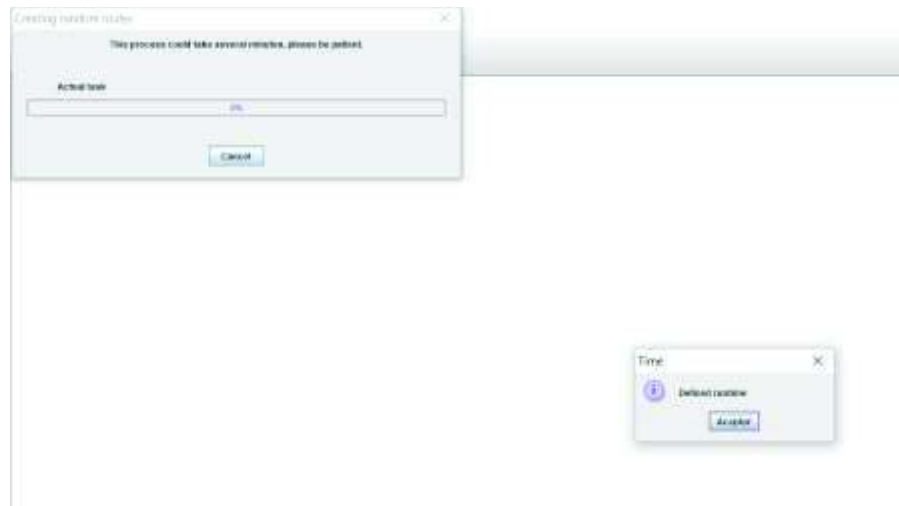


Figura 3.4 Generación de archivos para generacion simulación aleatorio



Figura 3.5 Simulación en programa SUMO para estudio

Tabla 3.1 Características y parámetros de vía 157005691

<way id="157005691" visible="true" version="13">
<nd ref="592079673"/>
<nd ref="1692394929"/>
<nd ref="1692394933"/>
<nd ref="1692394940"/>
<nd ref="1692394946"/>
<nd ref="1692394952"/>

<nd ref="1692394958"/>
<nd ref="3942012641"/>
<nd ref="1692394963"/>
<nd ref="1692394971"/>
<nd ref="5384908885"/>
<nd ref="5008036474"/>
<nd ref="1692394975"/>
<nd ref="5008036166"/>
<nd ref="5008036167"/>
<nd ref="1692394982"/>
<nd ref="1782672732"/>
<nd ref="1782672731"/>
<nd ref="1692394997"/>
<nd ref="1782672748"/>
<nd ref="1782672729"/>
<nd ref="1782672743"/>
<nd ref="1692395003"/>
<nd ref="1782672713"/>
<nd ref="1782672728"/>
<nd ref="1692395006"/>
<nd ref="1782672734"/>
<nd ref="1692395007"/>
<nd ref="1692395008"/>
<nd ref="1692395010"/>
<nd ref="5035490286"/>
<nd ref="1692394900"/>
<nd ref="5216470320"/>
<nd ref="1692395038"/>
<nd ref="1692395043"/>
<nd ref="1692395045"/>
<nd ref="1692395046"/>
<nd ref="1692395048"/>
<nd ref="5216470313"/>
<nd ref="1692395049"/>
<tag k="electrified" v="rail"/>
<tag k="frequency" v="0"/>
<tag k="gauge" v="1435"/>
<tag k="layer" v="-3"/>
<tag k="name" v="IND Eighth Avenue Line"/>
<tag k="railway" v="subway"/>
<tag k="tunnel" v="yes"/>
<tag k="voltage" v="625"/>
</way>

Tabla 3.2 Características y parámetros de vía 222299275

<way id="222299275" visible="true" version="12">
<nd ref="42428391"/>
<nd ref="4741363709"/>
<nd ref="4741363705"/>
<nd ref="42428385"/>
<nd ref="42428379"/>
<nd ref="42428377"/>
<nd ref="5010075967"/>
<nd ref="42428376"/>
<nd ref="5083818850"/>
<nd ref="42422042"/>
<nd ref="5083818851"/>
<nd ref="5083818846"/>
<nd ref="42428374"/>
<nd ref="5083818844"/>
<nd ref="5117245054"/>
<nd ref="42428373"/>
<nd ref="5083818841"/>
<nd ref="42428371"/>
<nd ref="5083818843"/>
<nd ref="42428370"/>
<nd ref="42428368"/>
<nd ref="42428366"/>
<nd ref="42428365"/>
<nd ref="4876543171"/>
<nd ref="42428362"/>
<nd ref="42428361"/>
<nd ref="42428355"/>
<nd ref="42428348"/>
<nd ref="42428346"/>
<nd ref="42428345"/>
<tag k="hgv" v="local"/>
<tag k="highway" v="secondary"/>
<tag k="lanes" v="3"/>
<tag k="lit" v="yes"/>
<tag k="maxspeed" v="25 mph"/>
<tag k="name" v="Broadway"/>
<tag k="name:en" v="Broadway"/>
<tag k="name:ru" v="Бродвей"/>
<tag k="oneway" v="yes"/>
<tag k="parking:lane:left" v="parallel"/>
<tag k="sidewalk" v="both"/>
<tag k="surface" v="asphalt"/>
<tag k="tiger:cfcc" v="A41"/>

<tag k="tiger:county" v="New York, NY"/>
<tag k="tiger:name_base" v="Broadway"/>
<tag k="tiger:zip_left" v="10003"/>
<tag k="tiger:zip_right" v="10003"/>
<tag k="wikidata" v="Q11260"/>
<tag k="wikipedia" v="en:Broadway (Manhattan)"/>
</way>

La primera vía id="157005691", es una vía que exclusivamente para transporte de tren, y por ende no debería circular ningún vehículo de esta simulación.

Esto se comprobó de dos maneras, la primera se lo hizo visualmente (en la simulación de SUMO 0.30.0) y la segunda mediante la lectura del archivo de rutas generado.

Este archivo de rutas generado se encuentra en su totalidad en el ANEXO 2, y se puede observar que la vía en cuestión (way id=157005691) no está considerada en ninguna posible ruta entre todas las rutas generadas aleatoriamente.

Cosa que no se da con la vía (way id="222299275") que es considerada en varias alternativas de rutas posibles que pueden tomar los vehículos, igualmente esta información se encuentra en el anexo anteriormente citado.

Adicionalmente al hacer un análisis de los parámetros en varios instantes determinados se tiene como resultado lo que se muestra en las figuras 3.6, 3.7 y 3.8. En donde se puede determinar que en la vía id="157005691" no se tiene ningún paso de vehículo a ningún tiempo. Para visualizar de manera más atractiva la información se la tabula en la tabla 3.3.

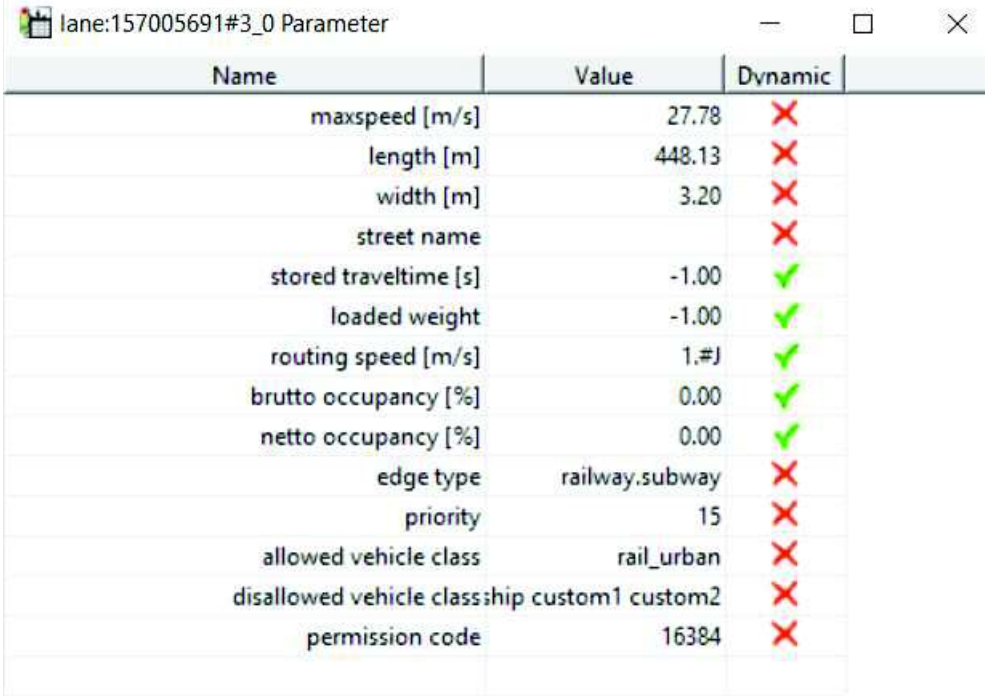
Tabla 3.3 Resultados de simulación vía 157005691 en el tiempo

Tiempo	Ocupación	Longitud	Velocidad Max
0	0	22.78	11.18 m/s
10	0	22.78	11.18 m/s
20	0	22.78	11.18 m/s
30	0	22.78	11.18 m/s
40	0	22.78	11.18 m/s
50	0	22.78	11.18 m/s
60	0	22.78	11.18 m/s
70	0	22.78	11.18 m/s
80	0	22.78	11.18 m/s
90	0	22.78	11.18 m/s
100	0	22.78	11.18 m/s

Como se debe suponer esto no sucede con la vía id="222299275" que evaluada en los mismos instantes de tiempo se demuestra que existe una diferente utilización según lo que se indica en la tabla 3.4 y las figuras 3.10 a 3.13

Tabla 3.4 Resultados de simulación vía 222299275 en el tiempo

Tiempo	Ocupación	Longitud	Velocidad Max
0	0	59.28	11.18 m/s
10	13.22	59.28	11.18 m/s
20	13.22	59.28	11.18 m/s
30	0	59.28	11.18 m/s
40	13.22	59.28	11.18 m/s
50	26.45	59.28	11.18 m/s
60	13.22	59.28	11.18 m/s
70	13.22	59.28	11.18 m/s
80	0	59.28	11.18 m/s
90	6.61	59.28	11.18 m/s
100	0	59.28	11.18 m/s



Name	Value	Dynamic
maxspeed [m/s]	27.78	✗
length [m]	448.13	✗
width [m]	3.20	✗
street name		✗
stored traveltime [s]	-1.00	✓
loaded weight	-1.00	✓
routing speed [m/s]	1.#J	✓
brutto occupancy [%]	0.00	✓
netto occupancy [%]	0.00	✓
edge type	railway.subway	✗
priority	15	✗
allowed vehicle class	rail_urban	✗
disallowed vehicle class:ship custom1 custom2		✗
permission code	16384	✗

Figura 3.6 Parámetros 157005691 vía tiempo 0

Name	Value	Dynamic
maxspeed [m/s]	27.78	✗
length [m]	448.13	✗
width [m]	3.20	✗
street name		✗
stored traveltime [s]	-1.00	✓
loaded weight	-1.00	✓
routing speed [m/s]	1.#J	✓
brutto occupancy [%]	0.00	✓
netto occupancy [%]	0.00	✓
edge type	railway.subway	✗
priority	15	✗
allowed vehicle class	rail_urban	✗
disallowed vehicle class	ship custom1 custom2	✗
permission code	16384	✗

Figura 3.8 Parámetros vía 157005691 tiempo 100

Name	Value	Dynamic
maxspeed [m/s]	11.18	✗
length [m]	59.28	✗
width [m]	3.20	✗
street name		✗
stored traveltime [s]	-1.00	✓
loaded weight	-1.00	✓
routing speed [m/s]	1.#J	✓
brutto occupancy [%]	0.00	✓
netto occupancy [%]	0.00	✓
edge type	highway.secondary	✗
priority	7	✗
allowed vehicle class	icicle custom1 custom2	✗
disallowed vehicle class	an rail rail_electric ship	✗
permission code	29237247	✗

Figura 3.9 Parámetros vía 222299277 tiempo 0

Name	Value	Dynamic
maxspeed [m/s]	11.18	✗
length [m]	56.72	✗
width [m]	3.20	✗
street name		✗
stored traveltime [s]	-1.00	✓
loaded weight	-1.00	✓
routing speed [m/s]	1.#J	✓
brutto occupancy [%]	13.22	✓
netto occupancy [%]	8.82	✓
edge type	highway.secondary	✗
priority	7	✗
allowed vehicle class	custom1 custom2	✗
disallowed vehicle class	an rail rail_electric ship	✗
permission code	29237247	✗

Figura 3.10 Parámetros vía 222299277 tiempo 50

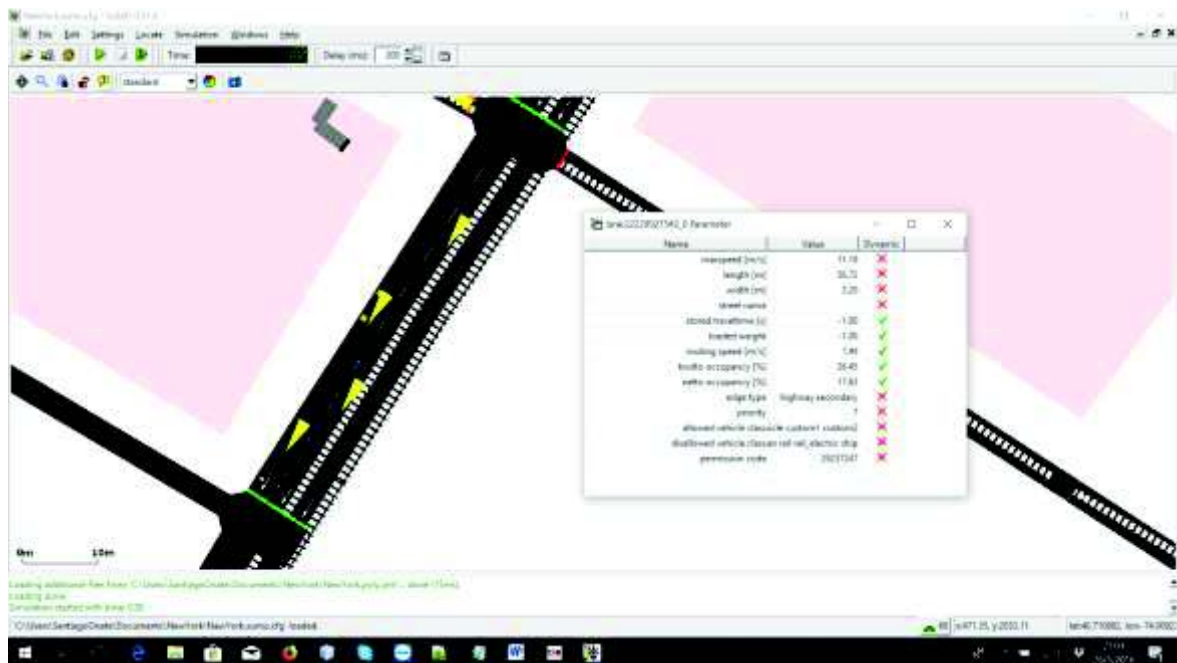


Figura 3.11 Parámetros vía 222299277 tiempo 100

Name	Value	Dynamic
maxspeed [m/s]	11.18	✗
length [m]	56.72	✗
width [m]	3.20	✗
street name		✗
stored traveltime [s]	-1.00	✓
loaded weight	-1.00	✓
routing speed [m/s]	1.#J	✓
brutto occupancy [%]	26.45	✓
netto occupancy [%]	17.63	✓
edge type	highway.secondary	✗
priority	7	✗
allowed vehicle class	icicle custom1 custom2	✗
disallowed vehicle class	an rail rail_electric ship	✗
permission code	29237247	✗

Figura 3.12 Parámetros vía 222299277 tiempo 100

Versión Original de C4R

Al realizar la simulación en el software original de C4R, visualmente es difícil distinguir el camino que toman los vehículos para la circulación, por cuanto la exportación al software de SUMO se lo hace en un solo color como se lo puede evidenciar en la figura 3.13.

Adicionalmente se puede caer en cuenta que no se puede distinguir los diferentes componentes del mapa real como son (tipos de vías, casas, ríos, camino peatonal, etc).

Pero al analizar el archivo generado de todas las posibles rutas que puede tomar los vehículos en la simulación se pudo determinar que se consideran todas las vías sin discriminar el tipo de vía que sea.

Es decir que las vías con id=157005691 e id=222299277 son consideradas en varias ocasiones, lo cual es correcto para la vía 222299277 pero no para la vía 157005691 por cuanto esta no puede ser ocupada por vehículos normales.

Este análisis comprueba que se está generando un error grave en la simulación y esto conlleva a que las simulaciones no entreguen resultados fiables.

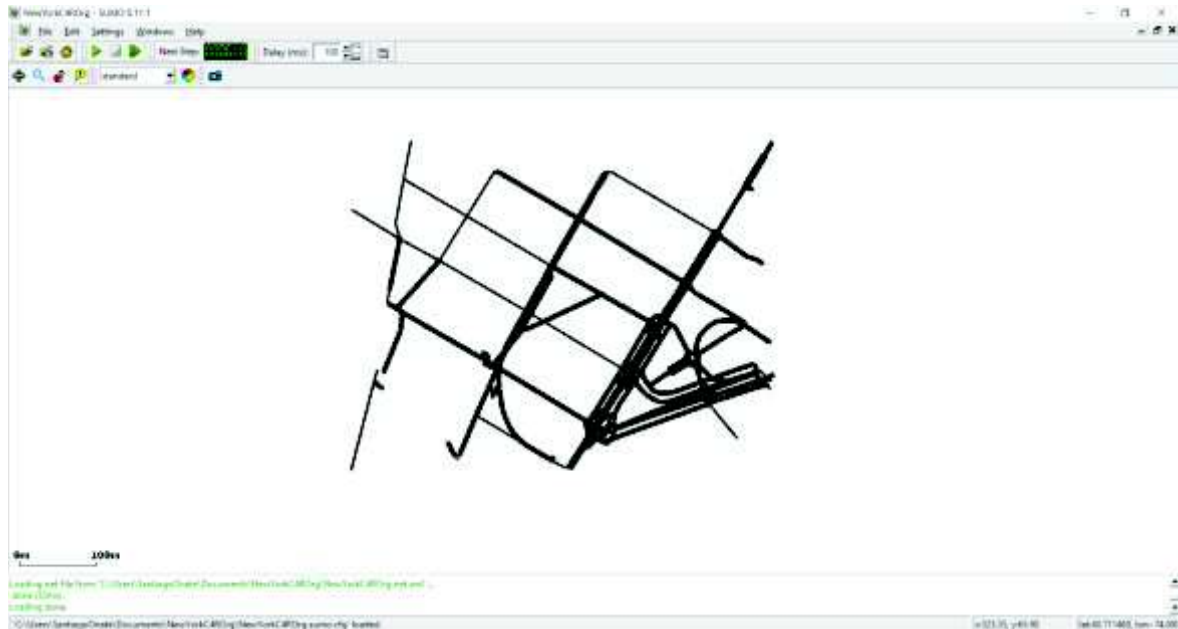


Figura 3.13 Simulación Original en SUMO version 0.11

3.2. Discusión

Como se pudo ver en las pruebas correspondientes lo que se hizo fue tratar de realizar la misma simulación bajo el mismo escenario, con las dos herramientas, la primera con la herramienta C4R mejorada y la segunda con la herramienta C4R Original.

Cabe recalcar que la importación de los mapas seleccionados para la simulación fue realizada con la misma herramienta JMapViewr, ya que está siendo ocupada en las dos versiones.

La diferencia se ve marcada en la manera de realizar la simulación. Por cuanto el software mejorado además de contar con los aplicativos Sumo, NetConvert y PolyConvert actualizados ocupa Scripts de Phytion para generar los archivos necesarios para la simulación.

Se pudo notar y comprobar que en la simulación realizada por el software C4R en su versión original no se discrimina las vías de tren o vías exclusivas de buses para enviar el tráfico de los vehículos simulados en el proyecto. Cosa que se lo puede deducir al observar en la tabla 3.3 en donde se puede apreciar que no existe un movimiento en la vía que no debería circular ningún vehículo.

Este error es corregido con el software C4R Mejorado ya que no en la simulación se nota que no circula ningún vehículo por las vías que no debería tener tráfico, esto se da por

cuanto intrínsecamente las herramientas actualizadas lo tienen y en el TypeMap de PolyConvert no se lo hace tomar en cuenta.

Ademas las herramientas actualizadas de Netconvert, Polyconvert y SUMO y los Scripts de Python hace que las simulaciones tengan un alto realismo, porque consideran factores externos como atenuación entre vehículos, aceleración, desaceleración, entre otros factores.

4. CONCLUSIONES

Para obtener simulaciones más realistas se necesitan que los parámetros de entrada sean apegados a la realidad, es por eso que si se consideran más atributos de entrada se podrán obtener resultados más confiables.

Es importante leer e investigar sobre las nuevas opciones que disponen las herramientas informáticas como por ejemplo SUMO, ya que siendo generado de un grupo de desarrollo, siempre están en constante estudio y liberan al mercado nuevas versiones que constan de mejoras en el proceso de simulación. Estas mejoras regularmente necesitan o contendrán información adicional que será necesario interpretarlas y manejarlas para un correcto uso. Todo esto ayudará sustancialmente a los resultados de las simulaciones.

Es importante tener las herramientas de software actualizadas (SUMO, NETCONVERT, POLYCONVERT y PYTHON) de tal manera que si se tiene una versión mejorada, se puedan analizar los cambios para mantener el correcto funcionamiento. Y así dar al programa resultante un uso más eficiente.

La herramienta POLYCONVERT es de mucha importancia para la representación en el entorno gráfico de SUMO por cuanto según los parámetros que se seleccionen se visualizará y el SUMO interpretará si se trata de una vía asignada para el tráfico o no, esto no se da en el software C4R original, ya que POLYCONVERT no existía para ese entonces.

En SUMO si se genera de manera incorrecta los componentes que intervienen en la simulación los resultados no van a ser fiables, es por eso que cuando se configure los parámetros que van a intervenir en la simulación como las vías para la generación de rutas, estas deben ser las que pueden ocuparse y por ende se debería descartar las que no se pueden utilizar; este proceso es realizado mediante el módulo mejorado de C4R en combinación con el Script de Python y una correcta discriminación en el archivo xml de OSM.

El considerar los parámetros correctos en la ejecución del comando de NETCONVERT ocasionaría que los resultados sean fiables para las simulaciones, ya que de este proceso se genera la ruta o red de la simulación, que al enlazar con el simulador de redes como ns2 (para la simulación de VANETs), resultará en tener datos apegados a la realidad.

Al tener una simulación casi real de los movimientos de los vehículos a lo largo de una red, se podrán simular las redes VANETs mediante el complemento de un simulador de red como ns2; que podrá asignar a los vehículos antenas, receptores, transmisores, atenuadores, repetidores. Para poder evaluar la calidad de la red generada y potencias de transmisión o pérdidas de señal a lo largo del tramo simulado a lo largo del tiempo.

Open Street Maps es un sistema muy usado y difundido en la cartografía mundial por cuanto es una herramienta Open Source. Cualquier cambio es difundido rápidamente y está siendo utilizado por diversos sectores de las telecomunicaciones de manera comercial e intelectual. Por estas razones es recomendable estar siempre al día con las actualizaciones y nuevos lanzamientos para poder adaptarlos a la herramienta C4R mejorada y no perder su funcionalidad.

Los archivos descargados mediante el Software de JMapView (herramienta que interactúa con el servidor de Open Street Maps) tienen una serie de parámetros e información que, si no se los sabe controlar o utilizar, interferirían en el resultado de una manera negativa. Es por eso que, mediante una correcta configuración de SUMO, una exitosa compilación y ejecución de NETCONVERT y POLYCONVERT, se podrán discriminar información no relevante y al realizar las simulaciones de tráfico vehicular los resultados van a ser altamente confiables.

Las pruebas de funcionamiento realizadas mediante el software original de C4R en su versión original con el SUMO 0.11, no discrimina las vías por donde los vehículos no pueden circular, por tanto arroja resultados que no son apegados a la realidad, esto es demostrado en las pruebas de funcionamiento que se las realizaron.

Existen una serie de opciones en la generación de archivos de rutas, estas nuevas opciones podrían considerarse para un nuevo estudio ya que lograrían generar más parámetros de entrada y por ende resultados más fiables. Una de estas opciones podría ser considerar el tipo de vía en relación a acabado físico de esta, ya que los vehículos preferirán viajar sobre vías asfaltadas mas no sobre lastradas o empedradas.

SUMO es una herramienta altamente poderosa para poder realizar simulaciones vehiculares, con las telecomunicaciones van tomando fuerza ya que sabiendo como se mueven los vehículos a lo largo de una región geográfica se podrá diseñar la manera de poder transmitir las señales entre estos de una manera más fiable, con bajos recursos energéticos y buenas velocidades de transmisión.

En el desarrollo de esta tesis se han ocupado 4 herramientas específicos que permitieron mejorar en un alto porcentaje la veracidad de los resultados de las simulaciones esta mejora se pudo cuantificar en un 40%, pero dentro de las librerías de SUMO existen mas programas que pueden ser utilizados que podrían mejorar aún más los resultados, sería recomendable hacer un estudio adicional con estas herramientas.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] Alberto Pardo Calvo, "C4R: Generación de modelos de movilidad para redes de vehículos a partir de mapas reales," 2011.
- [2] K. N. Cicenía Cárdena, "Uso de vehículos como sensores para estimar la intensidad de precipitación," 2016.
- [3] Domingo Aladrén Mari Carmen, "Simulación de VANETS (Vehicular Ad-Hoc Networks)," 2007.
- [4] S. – S. of U. MObility, A. Overview, and D. K. Michael Behrisch, Laura Bieker, Jakob Erdmann, "SUMO – Simulation of Urban MObility," *New York*, 2010.
- [5] O. O. Sarasti, "APLICACIONES PARA REDES VANET ENFOCADAS EN LA SOSTENIBILIDAD AMBIENTAL, UNA REVISIÓN SISTEMÁTICA," 2014.
- [6] Ripoll Cerezo, "Estudio del Simulador de Redes Vehiculares VEINS," 2012.
- [7] C. C. Shie Wang Yuan and Lin., *A Network Simulator for IEEE 802.11(p) and 1609 Wireless Vehicular Network*. 2010.
- [8] H. Hartenstein and K. Laberteaux, *VANET Vehicular Applications and Inter-Networking Technologies. Intelligent Transport Systems*. 2009.
- [9] M. Moustafa, H., Senouci, S. M., & Jerbi, *Introduction to Vehicular*. 2008.
- [10] G. Korkmaz, *Urban multi-hop broadcast protocol for inter-vehicle communication systems*. .
- [11] U. of Stuttgart, "CANU Mobility Simulation Environment (CanuMobiSim)," 2017. .
- [12] X. Zeng, R. Bagrodia, and M. Gerla, "GloMoSim: A Library for Parallel Simulation of Large-scale Wireless Networks," 2000.
- [13] M. Pachón, A., Nieto, C. & Velasco, "Modelos de Comportamiento de las Redes Vehiculares en sus Escenarios más Representativos, utilizando simulación en la herramienta NCTUns. Sistemas y Telemática," 2010.
- [14] "<http://web.scalable-networks.com/qualnet-network-simulator-software>," 2017. .
- [15] O. Foundation, "OSMs Open Street Maps," 2017, 2017.
- [16] S. Community, "Simulation of Urban MObility - Wiki," 2018.
- [17] S. Krauss, P. Wagner, and C. Gawron, *Metastable States in a Microscopic Model of*

Traffic Flow. 2007.

[18] S. Krauss, *Metastable States in a Microscopic Model of Traffic Flow*. 1998.

[19] P. Wagner, *How human drivers control their vehicle*. 2008.

6. ANEXOS

ANEXO I. Código fuente de Programa realizado.

ANEXO II. Archivo de rutas generado por el Software C4R Mejorado.

ANEXO III. Archivo de rutas generado por el Software C4R Original.

ANEXO I. CODIGO FUENTE DEL PROGRAMA

Este anexo muestra el código fuente de la programación realizada en NetBeans de Java, aquí se puede ver las clases, funciones y algoritmos para realizar los procesos mencionados en los capítulos dos y tres principalmente.

```
/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package c4r.gui;
import java.io.File;
import java.io.IOException;
import java.util.List;
import org.jdom2.Document;
import org.jdom2.Element;
import org.jdom2.JDOMException;
import org.jdom2.input.SAXBuilder;
import c4r.MobilityContainer;
import c4r.AppConfiguration;
import c4r.Scenario;
import c4r.enums.Extensions;
import c4r.items.network.SUMONetwork;
import c4r.items.route.RoutesContainer;
import com.teamdev.jxbrowser.chromium.Browser;
import com.teamdev.jxbrowser.chromium.BrowserType;
import com.teamdev.jxbrowser.chromium.swing.BrowserView;
import exceptions.SUMOException;
import io.PlainTextIO;
import io.XmlIO;
import java.awt.BorderLayout;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.PrintWriter;
import java.io.UnsupportedEncodingException;
import static java.lang.Integer.parseInt;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.HashSet;
import java.util.Set;
import java.util.logging.Level;
import java.util.logging.Logger;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.swing.JFrame;
import javax.swing.JInternalFrame;
import javax.swing.JOptionPane;
import static javax.swing.WindowConstants.DISPOSE_ON_CLOSE;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.transform.OutputKeys;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
```

```

import javax.xml.transform.stream.StreamResult;
import org.jdom2.Attribute;
import org.jdom2.filter.Filters;
import org.jdom2.output.Format;
import org.jdom2.output.XMLOutputter;
import org.jdom2.xpath.XPathExpression;
import org.jdom2.xpath.XPathFactory;
import org.w3c.dom.Node;
import simulation.TraceOptions;
import simulation.Visualization;
import utils.BufferCleaner;
import utils.SOValidator;
/**
 *
 * @author SantiagoO
 */
public class JFrameRoutes extends javax.swing.JFrame {

    public RoutesContainer container;
    private SUMONetwork network;
    private c4r.MobilityContainer mobcontainer;
    private String thepath;
    private Scenario scenario = null;
    private Workspace workspace = null;
    private String pathNetXML = "";
    private static final String WINDOWS_EXECUTABLE_0_31 = "\\sumo-
gui.exe";//Version 0.31.0;
    private static final String LINUX_EXECUTABLE = "sumo-gui";
    private ArrayList<String> listofways=new ArrayList<String>();
    private ArrayList<String> listWays=new ArrayList<String>();
    private ArrayList<String> listPossibleWays=new ArrayList<String>();
    private ArrayList<String> listOsmWays=new ArrayList<String>();
    String theFinalRoute;
    int depart=0;
    int newdepart=-0;
    int contadorprincial=0;
    /**
     * Creates new form JFrameRoutes
     */
    public JFrameRoutes(String path) {

        thepath=path;
        initComponents();
        fillcomboboxFirst();
        //Llenar el combobox con los modelos de movilidad
        cmbFollowModel.add("Choose one...");
        cmbFollowModel.add("Krauss");
        cmbFollowModel.add("KraussOrig1");
        cmbFollowModel.add("PWagner2009");
        cmbFollowModel.add("BKerner");
        cmbFollowModel.add("IDM");
        cmbFollowModel.add("IDMM");
        cmbFollowModel.add("KraussPS");
        cmbFollowModel.add("KraussAB");
        cmbFollowModel.add("SmartSK");
        cmbFollowModel.add("Wiedemann");
        cmbFollowModel.add("Daniell");
        //Llenar el combobox con los colores
        cmbColor.add("Choose one...");
        cmbColor.add("yellow");
    }
}

```



```

        cmbColor.add("red");
        cmbColor.add("green");
        cmbColor.add("blue");
        //Llenar el combobox con los tipos de Transporte
        cmbVehicleType.add("Choose one");
        cmbVehicleType.add("passenger");
        cmbVehicleType.add("taxi");
        cmbVehicleType.add("bus");
        cmbVehicleType.add("truck");
        cmbVehicleType.add("trailer");
        cmbVehicleType.add("rail");
        cmbVehicleType.add("rail_electric");
        cmbVehicleType.add("motorcycle");
        cmbVehicleType.add("moped");
        cmbVehicleType.add("bicycle");
        cmbVehicleType.add("pedestrian");
        cmbVehicleType.add("evehicle");
        cmbVehicleType.add("ship");
        //Llenar el combobox con el tiempo entre vehiculo
        cmbDepart.add("1");
        cmbDepart.add("5");
        cmbDepart.add("10");

    }

    /**
     * This method is called from within the constructor to initialize
     the form.
     * WARNING: Do NOT modify this code. The content of this method is
     always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jInternalFrame1 = new javax.swing.JInternalFrame();
        btnAdd = new javax.swing.JButton();
        btnCreateRoute = new javax.swing.JButton();
        label1 = new java.awt.Label();
        lstRoutes = new java.awt.List();
        cmbRoutes = new java.awt.Choice();
        label3 = new java.awt.Label();
        label4 = new java.awt.Label();
        label5 = new java.awt.Label();
        label6 = new java.awt.Label();
        label7 = new java.awt.Label();
        label8 = new java.awt.Label();
        txtAccel = new java.awt.TextField();
        txtSpeed = new java.awt.TextField();
        cmbFollowModel = new java.awt.Choice();
        cmbVehicleType = new java.awt.Choice();
        cmbColor = new java.awt.Choice();
        lstVehicle = new java.awt.List();
        btnRemoveCar = new javax.swing.JButton();
        btnAddCar = new javax.swing.JButton();
        label9 = new java.awt.Label();
        txtID = new java.awt.TextField();
        btnRemove = new javax.swing.JButton();
        lstRoutesCreated = new java.awt.List();

```

```

label2 = new java.awt.Label ();
btnRemoveRoute = new javax.swing.JButton ();
label10 = new java.awt.Label ();
btnAccept = new javax.swing.JButton ();
lstRoutesFinal = new java.awt.List ();
btnCreateRou = new javax.swing.JButton ();
txtDepart = new java.awt.TextField ();
label11 = new java.awt.Label ();
btnSimulation = new javax.swing.JButton ();
txtnCars = new java.awt.TextField ();
label12 = new java.awt.Label ();
cmbDepart = new java.awt.Choice ();
txtTotalCars = new java.awt.TextField ();
label13 = new java.awt.Label ();
btnReset = new javax.swing.JButton ();
lstRoutesName = new java.awt.List ();

setDefaultCloseOperation (javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

jInternalFrame1.setPreferredSize (new java.awt.Dimension (800,
800));
jInternalFrame1.setVisible (true);

javax.swing.GroupLayout jInternalFrame1Layout = new
javax.swing.GroupLayout (jInternalFrame1.getContentPane ());

jInternalFrame1.getContentPane ().setLayout (jInternalFrame1Layout);
jInternalFrame1Layout.setHorizontalGroup (

jInternalFrame1Layout.createParallelGroup (javax.swing.GroupLayout.Alignme
nt.LEADING)
    .addGap (0, 943, Short.MAX_VALUE)
);
jInternalFrame1Layout.setVerticalGroup (

jInternalFrame1Layout.createParallelGroup (javax.swing.GroupLayout.Alignme
nt.LEADING)
    .addGap (0, 1032, Short.MAX_VALUE)
);

btnAdd.setText ("ADD");
btnAdd.addActionListener (new java.awt.event.ActionListener () {
    public void actionPerformed (java.awt.event.ActionEvent evt) {
        btnAddActionPerformed (evt);
    }
});

btnCreateRoute.setText ("CREATE ROUTE");
btnCreateRoute.addActionListener (new
java.awt.event.ActionListener () {
    public void actionPerformed (java.awt.event.ActionEvent evt) {
        btnCreateRouteActionPerformed (evt);
    }
});

label11.setText ("Routes from OSM");

lstRoutes.setName ("lstListRoutes"); // NOI18N

```

```

cmbRoutes.setName("cmbRoutes"); // NOI18N
cmbRoutes.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        cmbRoutesMouseClicked(evt);
    }
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        cmbRoutesMouseEntered(evt);
    }
    public void mousePressed(java.awt.event.MouseEvent evt) {
        cmbRoutesMousePressed(evt);
    }
});
cmbRoutes.addItemListener(new java.awt.event.ItemListener() {
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
        cmbRoutesItemStateChanged(evt);
    }
});

label3.setText("Car definition");

label4.setText("Accel");

label5.setText("Speed");

label6.setText("Follow Model");

label7.setText("Vehicle Type");

label8.setText("Color");

txtAccel.setText("0.00");

txtSpeed.setText("0.00");

cmbFollowModel.setName("cmbFollowModel"); // NOI18N
cmbFollowModel.addMouseListener(new java.awt.event.MouseAdapter()
{
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        cmbFollowModelMouseClicked(evt);
    }
    public void mousePressed(java.awt.event.MouseEvent evt) {
        cmbFollowModelMousePressed(evt);
    }
});

cmbVehicleType.setName("cmbVehicleType"); // NOI18N
cmbVehicleType.addMouseListener(new java.awt.event.MouseAdapter()
{
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        cmbVehicleTypeMouseClicked(evt);
    }
    public void mousePressed(java.awt.event.MouseEvent evt) {
        cmbVehicleTypeMousePressed(evt);
    }
});

cmbColor.setName("cmbRoutes"); // NOI18N
cmbColor.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        cmbColorMouseClicked(evt);
    }
});

```

```

    }
    public void mousePressed(java.awt.event.MouseEvent evt) {
        cmbColorMousePressed(evt);
    }
});

lstVehicle.setName("lstVehicle"); // NOI18N
lstVehicle.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        lstVehicleActionPerformed(evt);
    }
});

btnRemoveCar.setText("REMOVE CAR");
btnRemoveCar.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnRemoveCarActionPerformed(evt);
    }
});

btnAddCar.setText("ADD CAR TYPE");
btnAddCar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnAddCarActionPerformed(evt);
    }
});

label9.setText("Id");

txtID.setText("Type0");

btnRemove.setText("REMOVE");
btnRemove.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnRemoveActionPerformed(evt);
    }
});

lstRoutesCreated.setName("lstListRoutes"); // NOI18N

label2.setText("Created Routes");

btnRemoveRoute.setText("REMOVE");
btnRemoveRoute.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnRemoveRouteActionPerformed(evt);
    }
});

label10.setText("Sumo File (rou.xml)");

btnAccept.setText("CREATE ROU.XML");
btnAccept.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnAcceptActionPerformed(evt);
    }
});

```

```

lstRoutesFinal.setName("lstListRoutes"); // NOI18N

btnCreateRou.setText("CREATE");
btnCreateRou.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCreateRouActionPerformed(evt);
    }
});

txtDepart.setText("Every");

label11.setText("Depart");

btnSimulation.setText("SIMULATION");
btnSimulation.setEnabled(false);
btnSimulation.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnSimulationActionPerformed(evt);
    }
});

txtnCars.setText("1");

label12.setText("Total Cars:");

cmbDepart.setName("cmbRoutes"); // NOI18N
cmbDepart.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        cmbDepartMouseClicked(evt);
    }
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        cmbDepartMouseEntered(evt);
    }
    public void mousePressed(java.awt.event.MouseEvent evt) {
        cmbDepartMousePressed(evt);
    }
});
cmbDepart.addItemListener(new java.awt.event.ItemListener() {
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
        cmbDepartItemStateChanged(evt);
    }
});

txtTotalCars.setEditable(false);
txtTotalCars.setEnabled(false);
txtTotalCars.setText("0");

label13.setText("# Cars");

btnReset.setText("RESET");
btnReset.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnResetActionPerformed(evt);
    }
});

lstRoutesName.setName("lstListRoutes"); // NOI18N

```

```

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout ( getContentPane () );
        getContentPane ().setLayout ( layout );
        layout.setHorizontalGroup (

layout.createParallelGroup ( javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup ( layout.createSequentialGroup ()
                .addContainerGap ()

.addGroup ( layout.createParallelGroup ( javax.swing.GroupLayout.Alignment.LE
ADING)
                .addGroup ( javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup ()
                        .addComponent ( label11,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap ( javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent ( txtDepart,
javax.swing.GroupLayout.PREFERRED_SIZE, 62,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap ( javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent ( cmbDepart,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addPreferredGap ( javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent ( label13,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap ( javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent ( txtnCars,
javax.swing.GroupLayout.PREFERRED_SIZE, 70,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addGap ( 69, 69, 69)
                                .addComponent ( btnCreateRou,
javax.swing.GroupLayout.PREFERRED_SIZE, 92,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                        .addComponent ( lstRoutesFinal,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                                .addGroup ( javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup ()

.addGroup ( layout.createParallelGroup ( javax.swing.GroupLayout.Alignment.LE
ADING)
                                .addGroup ( layout.createSequentialGroup ()
                                        .addComponent ( label1,
javax.swing.GroupLayout.PREFERRED_SIZE, 109,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap ( javax.swing.LayoutStyle.ComponentPlacement.RELATED, 21,
Short.MAX_VALUE)

```

```

        .addComponent (cmbRoutes,
javax.swing.GroupLayout.PREFERRED_SIZE, 302,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup (layout.createSequentialGroup ()
        .addComponent (lstRoutes,
javax.swing.GroupLayout.PREFERRED_SIZE, 158,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent (lstRoutesName,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED))

.addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent (btnAdd,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent (btnRemove,
javax.swing.GroupLayout.DEFAULT_SIZE, 101, Short.MAX_VALUE)
        .addComponent (btnReset,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 101, Short.MAX_VALUE)))
        .addGroup (javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup ()
        .addComponent (lstRoutesCreated,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent (btnRemoveRoute,
javax.swing.GroupLayout.PREFERRED_SIZE, 94,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap (30, 30, 30))
        .addComponent (lstVehicle,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGroup (layout.createSequentialGroup ()

.addGroup (layout.createSequentialGroup ()
        .addComponent (label5,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent (label3,
javax.swing.GroupLayout.PREFERRED_SIZE, 107,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent (label10,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent (btnCreateRoute)
        .addGroup (layout.createSequentialGroup ()

.addGroup (layout.createSequentialGroup ()
        .addComponent (label5,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent (label3,
javax.swing.GroupLayout.PREFERRED_SIZE, 107,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent (label10,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent (btnCreateRoute)
        .addGroup (layout.createSequentialGroup ()

```

```

                .addComponent (label4,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent (label9,
javax.swing.GroupLayout.PREFERRED_SIZE, 28,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap (11, 11, 11)

.addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING)

.addGroup (layout.createSequentialGroup ())

.addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING)

                .addComponent (txtID,
javax.swing.GroupLayout.PREFERRED_SIZE, 98,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING, false)

                .addComponent (txtAccel,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent (txtSpeed,
javax.swing.GroupLayout.PREFERRED_SIZE, 98,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addGap (30, 30, 30)

.addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING)

.addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.TRAILING, false)

                .addComponent (label7,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent (label6,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addComponent (label8,
javax.swing.GroupLayout.PREFERRED_SIZE, 52,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING, false)

                .addComponent (cmbVehicleType,
javax.swing.GroupLayout.DEFAULT_SIZE, 100, Short.MAX_VALUE)
                .addComponent (cmbColor,
javax.swing.GroupLayout.DEFAULT_SIZE, 100, Short.MAX_VALUE)
                .addComponent (cmbFollowModel,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
                .addComponent (btnAddCar,
javax.swing.GroupLayout.PREFERRED_SIZE, 176,
javax.swing.GroupLayout.PREFERRED_SIZE)))

```



```

        .addComponent(btnRemoveCar,
javax.swing.GroupLayout.PREFERRED_SIZE, 324,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(label2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(0, 0, Short.MAX_VALUE))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup())
        .addComponent(label12,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)

        .addComponent(txtTotalCars,
javax.swing.GroupLayout.PREFERRED_SIZE, 70,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGroup(layout.createSequentialGroup())
        .addComponent(btnAccept,
javax.swing.GroupLayout.PREFERRED_SIZE, 142,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(150, 150, 150)
        .addComponent(btnSimulation,
javax.swing.GroupLayout.PREFERRED_SIZE, 142,
javax.swing.GroupLayout.PREFERRED_SIZE))))))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jInternalFrame1,
javax.swing.GroupLayout.PREFERRED_SIZE, 959,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap())
);
layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup())

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)

        .addGroup(layout.createSequentialGroup())
        .addComponent(label3,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)

        .addGroup(layout.createSequentialGroup())

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)

        .addComponent(label9,
javax.swing.GroupLayout.PREFERRED_SIZE,

```

```

javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(txtID,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(2, 2, 2)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(label4,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(txtAccel,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(2, 2, 2)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(label5,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(txtSpeed,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup())

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(label6,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(cmbFollowModel,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(2, 2, 2)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(cmbVehicleType,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(label7,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(0, 0, 0)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(label8,
javax.swing.GroupLayout.PREFERRED_SIZE,

```

```

javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent (cmbColor,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent (btnAddCar)
            .addGap (2, 2, 2)

        .addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.TRAILING)
            .addGroup (layout.createSequentialGroup ()
                .addComponent (lstVehicle,
javax.swing.GroupLayout.PREFERRED_SIZE, 63,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap (2, 2, 2)
                .addComponent (btnRemoveCar)
                .addGap (27, 27, 27)

        .addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup (layout.createSequentialGroup ()

        .addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent (cmbRoutes,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent (label1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent (lstRoutes,
javax.swing.GroupLayout.PREFERRED_SIZE, 63,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGroup (layout.createSequentialGroup ()
            .addComponent (btnRemove)

        .addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent (btnReset))
            .addComponent (lstRoutesName,
javax.swing.GroupLayout.PREFERRED_SIZE, 63,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addComponent (btnAdd))

        .addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent (btnCreateRoute)

        .addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED)

```

```

        .addComponent (label2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent (lstRoutesCreated,
javax.swing.GroupLayout.PREFERRED_SIZE, 63,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent (btnRemoveRoute))
        .addGap (0, 0, 0)
        .addComponent (label10,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap (14, 14, 14)

.addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING)

.addGroup (layout.createSequentialGroup ()
        .addGap (3, 3, 3)

.addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent (label11,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent (txtDepart,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent (cmbDepart,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent (label13,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent (btnCreateRou))
        .addComponent (txtnCars,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent (lstRoutesFinal,
javax.swing.GroupLayout.PREFERRED_SIZE, 92,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED)

```

```

        .addComponent(txtTotalCars,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(label12,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(12, 12, 12)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
        .addComponent(btnAccept)
        .addComponent(btnSimulation)))
        .addComponent(jInternalFrame1,
javax.swing.GroupLayout.PREFERRED_SIZE, 1068,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(45, Short.MAX_VALUE))
);

label6.getAccessibleContext().setAccessibleName("");

pack();
} // </editor-fold>

private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
//Creación de una array de string para enviaar todas las
referencias
//que tiene una determinada vía (la seleccionada en el JComboBox
ArrayList<String> thereference=new ArrayList<String>();
String route;
String way;
String wayname;
//Extraer el string seleccionado en el JComboBox
route=cmbRoutes.getSelectedItem().toString();
//Solo el string del id de la vía
way=route.substring(route.lastIndexOf("/") + 1);
wayname=route.substring(0, route.lastIndexOf("/"));
//Se llama a la función que devuelve la via dada la referencia
thereference=waythenreferences(way,0);
//Borramos todo el contenido del combobox
cmbRoutes.removeAll();
//Funcion para llenar nuevamente el combobox y la lista de rutas
//creadas
doprocess(way);
//Llena la lista gráfica para la cracion de la ruta
lstRoutes.add(way);
lstRoutesName.add(wayname);
contadorprincial=0;
}

public ArrayList<String> waythenreferences(String theway, int pos)
{
//El path donde se encuentra todo el trabajo
String path=thepath;
//Código para leer archivos xml, e interpretarlos
SAXBuilder builder = new SAXBuilder();
File xmlFile = new File(path.substring(0, path.length() -
8)+"\osm");
//Creación de listas para leer los datos de las vías

```

```

ArrayList<String> thereference=new ArrayList<String>();
ArrayList<String> justthereference=new ArrayList<String>();
//Lectura de los archivos xml, y con opcion de "catch" errores
try {
    Document document = (Document) builder.build(xmlFile);
    Element rootNode = document.getRootElement();
    //Se crea una lista de todas las vías
    List childWay = rootNode.getChildren("way");
    for(int j=0;j<childWay.size();j++)
    {
        Element node = (Element) childWay.get(j);
        //Se crea una lista de todas las referencias de cada
vía
        List ndReference = node.getChildren("nd");

        if(node.getAttribute("id").toString().contains(theway))
            {
                for(int
item=0;item<ndReference.size();item++)
                {
                    Element reference = (Element)
ndReference.get(item);
                    String thereferenceok=theway+"-
"+reference.getAttribute("ref").toString().substring(17,
reference.getAttribute("ref").toString().length()-2);
                    thereference.add(thereferenceok);

justthereference.add(reference.getAttribute("ref").toString().substring(1
7,
reference.getAttribute("ref").toString().length()-2));
                }
            }
        }
    }
    catch (IOException io) {
        System.out.println(io.getMessage());
    } catch (JDOMException jdomex) {
        System.out.println(jdomex.getMessage());
    }
}
//Se retorna una lista con todas las referencias de la vía
return justthereference;
}

public ArrayList<String> referencetheways(String thereference, int
pos)
{
    String path=thepath;
    SAXBuilder builder = new SAXBuilder();
    File xmlFile = new File(path.substring(0, path.length() -
8)+"osm");
    ArrayList<String> theway=new ArrayList<String>();
    try {
        Document document = (Document) builder.build(xmlFile);
        Element rootNode = document.getRootElement();
        List childWay = rootNode.getChildren("way");
        for(int i=0;i<childWay.size();i++)
        {
            Element node = (Element) childWay.get(i);

```

```

        List ndReference = node.getChildren("nd");
        for(int j=0;j<ndReference.size();j++)
        {
            Element reference = (Element)
ndReference.get(j);

if(reference.getAttribute("ref").toString().contains(thereference))

theway.add(node.getAttribute("id").toString().substring(16,
node.getAttribute("id").toString().length()-2)+"-"+thereference);
        }
    }
    }
    catch (IOException io) {
        System.out.println(io.getMessage());
    } catch (JDOMException jdomex) {
        System.out.println(jdomex.getMessage());
    }
}
return theway;
}

private String findthename(String way)
{
    String osmwaytemp;
    String osmway="";
    for(int i=0;i<listOsmWays.size();i++)
    {
        osmwaytemp=listOsmWays.get(i);
        if(osmwaytemp.contains(way))
            osmway=osmwaytemp;
    }
    return osmway;
}

private void doprocess(String way)
{
    ArrayList<String> listRef=new ArrayList<String>();
    ArrayList<String> listWayTemp=new ArrayList<String>();
    ArrayList<String> listJustWayNoName=new ArrayList<String>();
    listRef=waythenreferences(way,0); //Tengo una lista con todas las
referencias
    for(int i=0;i<listRef.size();i++)
    {
        String laref=listRef.get(i);
        listWays=referencetheways(laref,0); //LLeno todas las vias
con una referencia i
        for (int j = 0; j < listWays.size(); j++)
            listWayTemp.add(listWays.get(j));
    }
    listofways=listWayTemp;
    String thename;
    for (int k = 0; k < listWayTemp.size(); k++)
    {
        int position1 = listWayTemp.get(k).indexOf("-");
        String newstring=listWayTemp.get(k);
        newstring=newstring.substring(0,position1);
        thename=findthename(newstring);
        listJustWayNoName.add(thename);
        Set<String> hs = new HashSet<String>();

```

```

        hs.addAll(listJustWayNoName);
        listJustWayNoName.clear();
        listJustWayNoName.addAll(hs);
    }
    for (int k = 0; k < listJustWayNoName.size(); k++)
        cmbRoutes.addItem(listJustWayNoName.get(k));
}

private ArrayList<String> hastheway(ArrayList<String> value, String
elid)
{
    ArrayList<String> listEdges=new ArrayList<String>();
    listEdges.clear();
    listEdges.add(elid);
    String path=thepath;
    SAXBuilder builder = new SAXBuilder();
    File xmlFile = new File(path.substring(0, path.length() -
8)+".rou.xml");
    try {
        Document document = (Document)
builder.build(xmlFile);
        Element rootNode = document.getRootElement();
        List childWay = rootNode.getChildren("vehicle");
        for(int j=0;j<childWay.size();j++)
        {
            Element node = (Element) childWay.get(j);
            List ndReference = node.getChildren("route");
            for(int
item=0;item<ndReference.size();item++)
            {
                String valor;
                Element reference = (Element)
ndReference.get(item);
                valor=reference.getAttribute("edges").toString();
                if(value.size()==1)
                {
                    listEdges.add(valor);
                }
                if(value.size()==2)
                {
                    String uno=value.get(1);
                    if(valor.contains(uno))
                        listEdges.add(valor);
                }
                if(value.size()==3)
                {
                    String uno=value.get(1);
                    String dos=value.get(2);
                    if(valor.contains(uno)&&
valor.contains(dos))
                        listEdges.add(valor);
                }
                if(value.size()==4)
                {
                    String uno=value.get(1);

```



```

        String dos=value.get(2);
        String tres=value.get(3);
        if(valor.contains(uno) &&
valor.contains(dos) && valor.contains(tres))
            listEdges.add(valor);
    }
    if(value.size()==5)
    {
        String uno=value.get(1);
        String dos=value.get(2);
        String tres=value.get(3);
        String cuatro=value.get(4);
        if(valor.contains(uno) &&
valor.contains(dos) && valor.contains(tres)
            && valor.contains(cuatro))
            listEdges.add(valor);
    }
    if(value.size()==6)
    {
        String uno=value.get(1);
        String dos=value.get(2);
        String tres=value.get(3);
        String cuatro=value.get(4);
        String cinco=value.get(5);
        if(valor.contains(uno) &&
valor.contains(dos) && valor.contains(tres)
            && valor.contains(cuatro) &&
valor.contains(cinco))
            listEdges.add(valor);
    }
    if(value.size()==7)
    {
        String uno=value.get(1);
        String dos=value.get(2);
        String tres=value.get(3);
        String cuatro=value.get(4);
        String cinco=value.get(5);
        String seis=value.get(6);
        if(valor.contains(uno) &&
valor.contains(dos) && valor.contains(tres)
            && valor.contains(cuatro) &&
valor.contains(cinco) && valor.contains(seis))
            listEdges.add(valor);
    }
    if(value.size()==8)
    {
        String uno=value.get(1);
        String dos=value.get(2);
        String tres=value.get(3);
        String cuatro=value.get(4);
        String cinco=value.get(5);
        String seis=value.get(6);
        String siete=value.get(7);
        if(valor.contains(uno) &&
valor.contains(dos) && valor.contains(tres)
            && valor.contains(cuatro) &&
valor.contains(cinco) && valor.contains(seis) && valor.contains(siete))
            listEdges.add(valor);
    }

```

```

    }
    }
}
catch (IOException io) {
    System.out.println(io.getMessage());
} catch (JDOMException jdomex) {
    System.out.println(jdomex.getMessage());
}

Collections.sort(listEdges, new Comparator<String>() {

    @Override
    public int compare(String o1, String o2) {
        if(o1.length()>o2.length()){
            return 1;
        }else{
            return o1.compareTo(o2);
        }
    }
});
return listEdges;
}

private ArrayList<String> findroute(int valor)
{
    ArrayList<String> list = null;
    String path=thepath;
    SAXBuilder builder = new SAXBuilder();
    File xmlFile = new File(path.substring(0, path.length() -
8)+"per.rou.xml");
    try {
        Document document = (Document) builder.build(xmlFile);
        Element rootNode = document.getRootElement();
        List nodoRoute = rootNode.getChildren("route");
        Element reference = (Element)
nodoRoute.get(valor);

        String
temp=reference.getAttribute("edges").toString();
        String
elid=reference.getAttribute("id").toString();
        String value;
temp=temp.substring(17, temp.length()-
3).substring(2);//antes -2

        String []strArray=temp.split(" ");
        for(int i=0; i<strArray.length;i++) {
            value=strArray[i];
            listPossibleWays.add(value);
        }
        list=hastheway(listPossibleWays,elid);
        listPossibleWays.clear();

    }
    catch (IOException io) {
        System.out.println(io.getMessage());
    } catch (JDOMException jdomex) {
        System.out.println(jdomex.getMessage());
    }
    return list;
}
}

```

```

private void cmbRoutesMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
}

private void cmbRoutesMousePressed(java.awt.event.MouseEvent evt) {
}

private void btnRemoveActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:

    lstRoutes.remove(lstRoutes.getSelectedIndex());
    int valor=lstRoutes.getSelectedIndex();
    lstRoutesName.remove(valor);
}

private int thenextnumber(int valor, int counter, int secondtime, int
counter1)
{
    if(secondtime==0)
    {
        counter=counter;
    }
    else if(secondtime==1)
    {
        counter=counter-1;
        valor=valor-1;
    }
    if(counter1==0)
    {
        valor=valor+counter+1;
    }
    else if(counter1==1)
    {
        valor=valor-counter-1;
    }
    return valor;
}
private String[] to(String way1, String way2, int count)
{
    String newfrom="";
    String newto="";
    //way1 es to
    //way2 es from
    int valor;//para el otro
    String elid;
    elid=way2.substring(way2.length()-1);
    valor=Integer.parseInt(elid);
    int secondtime=0;
    int flag=0;
    int counter=0;//estaba en 0
    int counter1=1;
    String path=thepath;
    String ways="";
    SAXBuilder builder = new SAXBuilder();
    //Leer el archivo de red generado

```

```

File xmlFile = new File(path.substring(0, path.length() -
8)+".net.xml");
try {
    Document document = (Document) builder.build(xmlFile);
    Element rootNode = document.getRootElement();
    //Hacer una lista con todos los elementos que tienen way
    List list = rootNode.getChildren("connection");
    int i=0;
    int value=list.size();
    String symb="";

do
{
    if(contadorprincial==0)
    {
        Element node = (Element) list.get(i);
        //Hacer una lista con todos los elementos que
        tienen to
        String to;

to=node.getAttribute("to").toString().substring(16).substring(0,node.getA
ttribute("to").toString().substring(16).length()-2);
        String from;

from=node.getAttribute("from").toString().substring(18).substring(0,node.
getAttribute("from").toString().substring(18).length()-2);
        if(to.contains(symb+way1) && from.contains(way2))
        {
            newfrom = from;//.substring(0, from.length()
- 2);
            newto = to;//.substring(0, to.length() - 2);
            ways=from.substring(0, from.length() - 2) + "
" + to.substring(0, to.length() - 2);;
            flag=1;
        }
        else
        {
            newfrom = "";
            newto = "";
            ways="";
            flag=0;
        }
        i++;
    if(i==list.size())
    {
        String theorgfrom;
        theorgfrom=way2;
        String numerostring;
        String signo;
        String cantidad;
        int thenextnumber;

thenextnumber=thenextnumber(valor,counter,secondtime,counter1);
        valor=thenextnumber;
        counter++;
        numerostring=String.valueOf(valor);
        String thenewto;
        if(valor==0)
            cantidad="0";
    }
}
}

```

```

        if(numerostring.startsWith("-"))
        {
            //Para controlar los que no son -
            if(way2.contains("-"))
                thenewto=way2.substring(1,
way2.length()-1);
            else
                thenewto=way2.substring(0,
way2.length()-1);
            signo="-";
        }
        cantidad=numerostring.substring(1,numerostring.length());
    }
    else
    {
        thenewto=way2.substring(0, way2.length()-
1);
        signo="";
        cantidad=numerostring;
        thenewto=way2.substring(0, way2.length()-
1);
    }
    if(valor<-30)
    {
        way2=theorgfrom;
        if(way2.contains("-"))
            way1=way2.substring(1,way2.length());
        else
            way1=way2.substring(0,way2.length());
    }
    else
    {
        way1=signo+thenewto+cantidad;
        way2=theorgfrom;
    }
    i=0;
    counter1=0;
    if((counter/2)% 2 == 0)
        secondtime=0;
    else
        secondtime=1;
}
if(valor>1000)
{
    flag=1;
    cmbRoutes.getItem(i);
    JOptionPane.showMessageDialog(null, "Wrong Way
in: " + cmbRoutes.getItem(i));
    lstRoutes.removeAll();
    cmbRoutes.removeAll();
    fillcomboboxFirst();
}
}

else
{
    Element node = (Element) list.get(i);

```

```

//Hacer una lista con todos los elementos que
tienen to
String to;

to=node.getAttribute("to").toString().substring(16).substring(0,node.getA
ttribute("to").toString().substring(16).length()-2);
String from;

from=node.getAttribute("from").toString().substring(18).substring(0,node.
getAttribute("from").toString().substring(18).length()-2);
    if(to.contains(symb+way1) &&from.equals(way2))
    {
        newfrom = from;
        newto = to;
        ways=from.substring(0, from.length() - 2) + "
" + to.substring(0, to.length() - 2);;
        flag=1;
    }
    else
    {
        newfrom ="";
        newto = "";
        ways="";
        flag=0;
    }
    i++;

if(i==list.size())
{
    String theorgfrom;
    theorgfrom=way2;
    String numerostring;
    String signo;
    String cantidad;
    int thenextnumber;

thenextnumber=thenextnumber(valor,counter,secondtime,counter1);
    valor=thenextnumber;
    counter++;
    numerostring=String.valueOf(valor);
    String thenewto;
    if(valor==0)
        cantidad="0";
    if(numerostring.startsWith("-"))
    {
        //Para controlar los que no son -
        if(way2.contains("-"))
            thenewto=way2.substring(1,
way2.length()-1);

        else
            thenewto=way2.substring(0,
way2.length()-1);

        signo="-";

    cantidad=numerostring.substring(1,numerostring.length());
    }
    else
    {
        thenewto=way2.substring(0, way2.length()-
1);

```

```

        signo="";
        cantidad=numerostring;
        thenewto=way2.substring(0, way2.length()-
1);
    }
    if(valor<-30)
    {
        way2=theorgfrom;
        if(way2.contains("-"))
            way1=way2.substring(1,way2.length());
        else
            way1=way2.substring(0,way2.length());
    }
    else
    {
        way1=signo+thenewto+cantidad;
        way2=theorgfrom;
    }
    i=0;
    counter1=0;
    if((counter/2)% 2 == 0)
        secondtime=0;
    else
        secondtime=1;
}
if(valor>1000)
{
    flag=1;
    JOptionPane.showMessageDialog(null, "Wrong Way
in: " + cmbRoutes.getItem(i));
    lstRoutes.removeAll();
    cmbRoutes.removeAll();
    fillcomboboxFirst();
}
}

}while(flag!=1);
contadorprincial=1;
}
catch (IOException io) {
    System.out.println(io.getMessage());
} catch (JDOMException jdomex) {
    System.out.println(jdomex.getMessage());
}
return new String[] {newfrom, newto,
Integer.toString(counter)};
}

private void btnCreateRouteActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:

    int contador;
    contador=0;
    String theString;
    String prueba="";
    String substring="";

```

```

        theString="route id="+\""+\"route\" +
String.valueOf(lstRoutesCreated.getItemCount()) +\""+\" edges=\" + \"\";
        int thecounter=lstRoutes.getItemCount();
        do
        {
            String theway="";
            String theto=lstRoutes.getItem(contador+1);
            String thefrom=lstRoutes.getItem(contador);
            String result[] = to(theto,thefrom,contador);
            String newfrom;
            newfrom=result[0];
            String newto;
            newto=result[1];
            String other;
            other=result[2];
            int flag2;
            flag2=Integer.parseInt(other);
            String thewaysneed="";
            thewaysneed=newfrom;
            if(flag2!=0)
            {
                int k=lstRoutes.getItemCount();
                lstRoutes.add(lstRoutes.getItem(k-1), k);
                do
                {
                    lstRoutes.replaceItem(lstRoutes.getItem(k-2), k-1);
                    k--;
                }while(contador<k);
                lstRoutes.replaceItem(newfrom, contador);
                lstRoutes.replaceItem(newto, contador+1);
                thecounter--;
                contador ++;
            }while(contador<lstRoutes.getItemCount()-1);

            for(int j=0;j<lstRoutes.getItemCount();j++)
            {
                prueba=prueba+" "+lstRoutes.getItem(j);
            }
            lstRoutesCreated.add(theString+prueba+ \"\");
            lstRoutes.removeAll();
            lstRoutesName.removeAll();
            cmbRoutes.removeAll();
            fillcomboBoxFirst();

        }

        private void cmbFollowModelMouseClicked(java.awt.event.MouseEvent
evt) {
            // TODO add your handling code here:
        }

        private void cmbFollowModelMousePressed(java.awt.event.MouseEvent
evt) {
            // TODO add your handling code here:
        }

        private void cmbVehicleTypeMouseClicked(java.awt.event.MouseEvent
evt) {

```



```

        // TODO add your handling code here:
    }

    private void cmbVehicleTypeMousePressed(java.awt.event.MouseEvent
    evt) {
        // TODO add your handling code here:
    }

    private void cmbColorMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
    }

    private void cmbColorMousePressed(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
    }

    private void btnRemoveCarActionPerformed(java.awt.event.ActionEvent
    evt) {
        // TODO add your handling code here:
        lstVehicle.remove(lstVehicle.getSelectedItem());
    }

    private void btnAddCarActionPerformed(java.awt.event.ActionEvent evt)
    {
        // TODO add your handling code here:
        if(cmbFollowModel.getSelectedItem().toString()=="Choose one...")
        {
            JOptionPane.showMessageDialog(null, "You must choose a Follow
    Model");
            return;
        }
        if(cmbVehicleType.getSelectedItem().toString()=="Choose one...")
        {
            JOptionPane.showMessageDialog(null, "You must choose");
            return;
        }
        if(cmbColor.getSelectedItem().toString()=="Choose one...")
        {
            JOptionPane.showMessageDialog(null, "You must choose a
    Color");
            return;
        }
        lstVehicle.add("vType id=" + "\"" + txtID.getText() + "\" "
            + "accel=" + "\"" + txtAccel.getText() + "\" "
            + "speed=" + "\"" + txtSpeed.getText() + "\" "
            + "carFollowModel=" + "\"" +
    cmbFollowModel.getSelectedItem().toString() + "\" "
            + "vClass=" + "\"" +
    cmbVehicleType.getSelectedItem().toString() + "\" "
            + "color=" + "\"" + cmbColor.getSelectedItem().toString()
    + "\"");
        txtID.setText("Type"+lstVehicle.getItemCount());
    }

    private void btnRemoveRouteActionPerformed(java.awt.event.ActionEvent
    evt) {
        // TODO add your handling code here:
        lstRoutesCreated.remove(lstRoutesCreated.getSelectedItem());
    }

```

```

private void makexml ()
{
    ArrayList<String> listofedges=new ArrayList<String>();

    PrintWriter writer2 = null;
    try{

        // TODO add your handling code here:
        writer2 = new PrintWriter(thepath.substring(0,
thepath.length() - 8) + "per2.rou.xml", "UTF-8");
        writer2.println("<?xml version=\"1.0\"?>");
        writer2.println("<routes>");
        int countvehicle=0;
        do{
            String value;
            value=lstVehicle.getItem(countvehicle).toString();
            writer2.println("<"+value+\"/>");
            countvehicle++;
        }while(countvehicle<lstVehicle.getItemCount());
        int countroutes=0;
        do{

            listofedges=findroute(countroutes);
            String nuevoedge;
            String elid;

            elid=listofedges.get(0);//el primer valor
            elid=elid.substring(12, elid.length()-2);
            nuevoedge= listofedges.get(1);//el primer
valor

            nuevoedge=nuevoedge.toString().substring(19,
nuevoedge.length()-2);

            writer2.println("<route "+elid+"\"
edges=\""+nuevoedge+"\"+\"/>");
            countroutes++;
        }while(countroutes<lstRoutesCreated.getItemCount());
        int countsimulation=0;
        do{

            writer2.println("<"+lstRoutesFinal.getItem(countsimulation).toString()+\"
>");

            countsimulation++;
        }while(countsimulation<lstRoutesFinal.getItemCount());
        writer2.println("</routes>");

    } catch (FileNotFoundException ex) {

        Logger.getLogger(JFrameRoutes.class.getName()).log(Level.SEVERE, null,
ex);

    } catch (UnsupportedEncodingException ex) {
        //JOptionPane.showMessageDialog(null, "Route not valid");

        Logger.getLogger(JFrameRoutes.class.getName()).log(Level.SEVERE, null,
ex);
    } finally {
        //JOptionPane.showMessageDialog(null, "Route not valid");
        btnSimulation.setEnabled(true);
    }
}

```

```

        writer2.close();
    }

}

private void btnAcceptActionPerformed(java.awt.event.ActionEvent evt)
{

    PrintWriter writer2 = null;
    try{

        // TODO add your handling code here:
        writer2 = new PrintWriter(thepath.substring(0,
thepath.length() - 8) + "per.rou.xml", "UTF-8");
        writer2.println("<?xml version=\"1.0\"?>");
        writer2.println("<routes>");
        int countvehicle=0;
        do{

writer2.println("<"+lstVehicle.getItem(countvehicle).toString()+"/>");
            countvehicle++;
        }while(countvehicle<lstVehicle.getItemCount());
        int countroutes=0;
        do{

writer2.println("<"+lstRoutesCreated.getItem(countroutes)+"/>");
            countroutes++;
        }while(countroutes<lstRoutesCreated.getItemCount());
        int countsimulation=0;
        do{

writer2.println("<"+lstRoutesFinal.getItem(countsimulation).toString()+"/
>");
            countsimulation++;
        }while(countsimulation<lstRoutesFinal.getItemCount());
        writer2.println("</routes>");

        } catch (FileNotFoundException ex) {

Logger.getLogger(JFrameRoutes.class.getName()).log(Level.SEVERE, null,
ex);

        } catch (UnsupportedEncodingException ex) {

Logger.getLogger(JFrameRoutes.class.getName()).log(Level.SEVERE, null,
ex);

        } finally {
            writer2.close();
        }
        makexml();

    }

private void btnCreateRouActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:
    String thestring;

```

```

        int value;

value=Integer.parseInt(txtnCars.getText())+Integer.parseInt(txtTotalCars.
getText());
    txtTotalCars.setText(Integer.toString(value));
    int ncars;
    int values=0;
    ncars=Integer.parseInt(txtnCars.getText());
    newdepart=depart;
    do
    {

depart=newdepart+Integer.parseInt(cmbDepart.getSelectedItem())*values;
    thestring="vehicle
id="+"\\""+lstRoutesFinal.getItemCount()+"\" type="+"\\""+
lstVehicle.getSelectedItem().toString().substring(10,15)+"\" +
"
route="+"\\""+lstRoutesCreated.getSelectedItem().toString().substring(10,1
6)+"\""+
        //" depart="+"\\""+txtDepart.getText()+"\"";
        " depart="+"\\""+depart+"\\"";
    lstRoutesFinal.add(thestring);
    values++;
    }while(values<ncars);
}

    public static void visualizeWithSUMO(String configuration) throws
FileNotFoundException, IOException, InterruptedException, SUMOException {
    final String cmd = " -c " + configuration; //Version 0.11.1
    String execution = "";

    if (SOValidator.isWindows()) {
        AppConfiguraton conf = new AppConfiguraton();
        conf.load();
        execution = conf.getSumoPath() + WINDOWS_EXECUTABLE_0_31 +
cmd;
    } else if (SOValidator.isUnix()) {
        execution = LINUX_EXECUTABLE + cmd; //SUMO v0.12.2
    }
    Process p = Runtime.getRuntime().exec(execution);
    /*Cleans the output and error buffers to avoid the process
hangs.*/
    BufferCleaner bcOut = new BufferCleaner(p.getInputStream());
    BufferCleaner bcError = new BufferCleaner(p.getErrorStream());
    bcOut.start();
    bcError.start();
    int outValue = p.waitFor();
    if (outValue != 0) {
        System.out.println("log::"+bcError.getOutput());
        throw new SUMOException("Impossible visualize the
simulation.\n");
    }
}
}

```

```

public void visualizeSimulation() {
    try {
        if (!new File(scenario.getWorkingPath() + "random_vehicles" +
Extensions.ROUTE.getExtension()).exists()) {
            ProgressDialog pd = new ProgressDialog(this, "Creating
random routes",
                "This process could take several minutes, please
be patient.",
                scenario,
                false);
            pd.setVisible(true);
        }
        Thread t = new Thread(new Runnable() {

            @Override
            public void run() {
                try {
                    String configurationFile =
thepath.replace("c4r.xml", "sumo.cfg");
                    exportFile(configurationFile);

Visualization.visualizeWithSUMO(configurationFile);
                } catch (Exception ex) {
                    JOptionPane.showMessageDialog(null,
ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);

Logger.getLogger(MainWindow.class.getName()).log(Level.SEVERE, null, ex);
                }
            }
        });
        t.start();
        workspace.repaint();
    } catch (NullPointerException e) {
        JOptionPane.showMessageDialog(this, "The mobility model is
not created yet.",
            "Visualization error",
            JOptionPane.ERROR_MESSAGE);
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, e.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
    }
}

public void exportFile(String name) throws Exception {

    try {
        org.w3c.dom.Document doc = XmlIO.createEmptyXML();
        org.w3c.dom.Element root =
doc.createElement("configuration");
        doc.appendChild(root);
        org.w3c.dom.Element input = doc.createElement("input");
        root.appendChild(input);
        org.w3c.dom.Element net_file = doc.createElement("net-file");
        //net_file.setAttribute("value", network.getPath());
        String netvalue;
        netvalue=thepath.replace("c4r.xml", "net.xml");
        net_file.setAttribute("value", netvalue);
        input.appendChild(net_file);
    }
}

```

```

        org.w3c.dom.Element route_files = doc.createElement("route-
files");
        String routeFiles = "";
        routeFiles = name.substring(0, name.length()-
9).concat(Extensions.ROUTE.getExtension());
        String routevalue;
        routevalue=routeFiles.replace(".rou","per.rou");
        route_files.setAttribute("value", routevalue );
        input.appendChild(route_files);
        String polyFile = name.substring(0, name.length()-
9).concat(Extensions.POLY.getExtension());
        org.w3c.dom.Element additional_file =
doc.createElement("additional-files");
        additional_file.setAttribute("value", polyFile);
        input.appendChild(additional_file);
        org.w3c.dom.Element time = doc.createElement("time");
        root.appendChild(time);
        org.w3c.dom.Element startTime = doc.createElement("begin");
        startTime.setAttribute("value", "0");
        time.appendChild(startTime);
        XmlIO.save(doc, name);
    }
    catch(Exception e) {
        throw e;
    }
}

private void btnSimulationActionPerformed(java.awt.event.ActionEvent
evt) {

    try {
        parseToSUMONet ();
    } catch (IOException ex) {

Logger.getLogger(JFrameRoutes.class.getName()).log(Level.SEVERE, null,
ex);
        } catch (InterruptedException ex) {

Logger.getLogger(JFrameRoutes.class.getName()).log(Level.SEVERE, null,
ex);
        } catch (SUMOException ex) {

Logger.getLogger(JFrameRoutes.class.getName()).log(Level.SEVERE, null,
ex);
        }
        int valor=Integer.parseInt(txtTotalCars.getText ());
        try {
            ejecutarComandoPolyConvertPersonal(valor);
        } catch (IOException ex) {

Logger.getLogger(JFrameRoutes.class.getName()).log(Level.SEVERE, null,
ex);
        } catch (InterruptedException ex) {

Logger.getLogger(JFrameRoutes.class.getName()).log(Level.SEVERE, null,
ex);
        } catch (SUMOException ex) {

Logger.getLogger(JFrameRoutes.class.getName()).log(Level.SEVERE, null,
ex);
        }
}

```

```

    }
    String configurationFile = thepath.replace("c4r.xml","sumo.cfg");
    try {
        exportFile(configurationFile);
    } catch (Exception ex) {

Logger.getLogger(JFrameRoutes.class.getName()).log(Level.SEVERE, null,
ex);
    }
    try {
        visualizeWithSUMO(configurationFile);
    } catch (IOException ex) {

Logger.getLogger(JFrameRoutes.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (InterruptedException ex) {

Logger.getLogger(JFrameRoutes.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (SUMOException ex) {

Logger.getLogger(JFrameRoutes.class.getName()).log(Level.SEVERE, null,
ex);
    }

    }

    private void lstVehicleActionPerformed(java.awt.event.ActionEvent
    evt) {
        // TODO add your handling code here:
    }

    private void cmbRoutesMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
    }

    private void cmbRoutesItemStateChanged(java.awt.event.ItemEvent evt)
    {
        // TODO add your handling code here:
        Browser browser = new Browser(BrowserType.LIGHTWEIGHT);
        BrowserView view = new BrowserView(browser);
        String way;
        way=cmbRoutes.getSelectedItem().toString();
        int valor;
        int position = way.indexOf("/");
        String theway;
        theway=way.substring(position+1);
        browser.loadURL("https://www.openstreetmap.org/way/"+theway);
        jInternalFrame1.setContentPane(view);
        jInternalFrame1.setLocation(100 , 100 );
        jInternalFrame1.setSize(800, 800);
        jInternalFrame1.setVisible(true);
    }

    private void cmbDepartMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
    }

```

```

private void cmbDepartMouseEntered(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
}

private void cmbDepartMousePressed(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
}

private void cmbDepartItemStateChanged(java.awt.event.ItemEvent evt)
{
    // TODO add your handling code here:
}

private void btnResetActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    lstRoutes.removeAll();
    lstRoutesName.removeAll();
    cmbRoutes.removeAll();
    lstRoutesCreated.removeAll();
    fillcomboBoxFirst();
    contadorprincial=0;
}

public ArrayList<String> referencethenwaysAll(String thereference,
int pos)
{
    String path=thepath;
    SAXBuilder builder = new SAXBuilder();
    File xmlFile = new File(path.substring(0, path.length() -
8)+".osm");
    ArrayList<String> theway=new ArrayList<String>();
    try {
        Document document = (Document) builder.build(xmlFile);
        Element rootNode = document.getRootElement();
        List childWay = rootNode.getChildren("way");
        for(int i=0;i<childWay.size();i++)
        {
            Element node = (Element) childWay.get(i);
            List ndReference = node.getChildren("nd");
            for(int j=0;j<ndReference.size();j++)
            {
                Element reference = (Element)
ndReference.get(j);

                if(reference.getAttribute("ref").toString().contains(thereference))
                    //theway.add(theway+"-
"+reference.getAttribute("ref").toString().substring(17,
reference.getAttribute("ref").toString().length()-2));

                theway.add(node.getAttribute("id").toString().substring(16,
node.getAttribute("id").toString().length()-2)+"-"+thereference);
            }
        }
    }
    catch (IOException io) {

```



```

        System.out.println(io.getMessage());
    } catch (JDOMException jdomex) {
        System.out.println(jdomex.getMessage());
    }
}
return theway;
}

public MyResult information(int value, List listtag, int theelement)
{
    //Inicialización de variables con valores aleatorios
    int havename=-1;
    int position=-1;
    int notway=-1;
    MyResult theresult=new MyResult();
    int thesize;
    thesize=value;
    int valor=0;
    int i=0;
    if(thesize>0)
    {
        do
        {
            String nodotag;
            Element tag = (Element) listtag.get(i);
            nodotag=tag.getAttribute("k").toString();
            if(nodotag.contentEquals("[Attribute: k=\"name\"]")
                &&!nodotag.contains("building")
                &&!nodotag.contains("train")
                &&!nodotag.contains("train")
                &&!nodotag.contains("pedestrian")
                &&!nodotag.contains("path")
                &&!nodotag.contains("bridleway")
                &&!nodotag.contains("cycleway")
                &&!nodotag.contains("footway"))//si tiene el el
nombre de la vía

                {
                    havename=1;//Si tiene nombre de la vía
                    position=i ;//Posicion del elemento tag que tiene el
nombre--

                    notway=0;//si es una vía para vehículo
                    valor=44548;
                }
            //Discriminar que no se utlice las vías y edificios en las
            //simulaciones
            else
            {
                havename=0;//No tiene nombre de la vía
                position=i;//Posición
                notway=0;//Si es una vía para simulación
                valor=321;
            }
            value--;
            i++;
        }while (valor!=44548&&value!=0);
    }
    //Enviar los resultados para ser ocupados
    theresult.havename=havename;
    theresult.position=position;
    theresult.notway=notway;
}

```

```

        //Devolver la variable para escritura de resultados
        return theresult;
    }

public MyResult TheVelocity(int value, List listtag, int theelement)
{
    //Inicialización de variables con valores aleatorios
    int havename=-1;
    int position=-1;
    int notway=-1;
    MyResult theresult=new MyResult();
    int thesize;
    thesize=value;
    int valor=0;
    int i=0;
    if(thesize>0)
    {
        do
        {
            String nodotag;
            Element tag = (Element) listtag.get(i);
            nodotag=tag.getAttribute("k").toString();
            if(nodotag.contentEquals("[Attribute: k=\"highway\"]"))

            {
                havename=1;//Si tiene nombre de la vía
                position=i ;//Posicion del elemento tag que tiene el
nombre--

                notway=0;//si es una vía para vehiculo
                valor=44548;
            }
            //Discriminar que no se utlice las vías y edificios en las
            //simulaciones
            else
            {
                havename=0;//No tiene nombre de la vía
                position=i;//Posición
                notway=0;//Si es una vía para simulación
                valor=321;
            }
            value--;
            i++;
        }while (valor!=44548&&value!=0);
    }
    //Enviar los resultados para ser ocupados
    theresult.havename=havename;
    theresult.position=position;
    theresult.notway=notway;
    //Devolver la variable para escritura de resultados
    return theresult;
}

public ArrayList<String> info(int value, List listtag)
{
    ArrayList<String> info=new ArrayList<String>();
    String havename="0";
    info.add(havename);
    info.add("2");
    for(int i=0;i<listtag.size();i++)
    {

```

```

String nodotag;
Element tag = (Element) listtag.get(value);
nodotag=tag.getAttribute("k").toString();
if(nodotag.contains("name"))
{
    info.clear();
    havename="1";
    info.add(havename);
    info.add(Integer.toString(i));

}
else
if(nodotag.contains("building")||nodotag.contains("train")||nodotag.contains("train"))

||nodotag.contains("pedestrian")||nodotag.contains("path")||nodotag.contains("bridleway")

||nodotag.contains("cycleway")||nodotag.contains("footway"))
{
    info.add("101010");
}
value--;
}

return info;
}

public void fillcomboboxFirst()
{
    //Clase creada para discriminar si la via tiene o no nombre
    //y ademas si si puede moverse un vehiculo por esta
    MyResult info=new MyResult();// ArrayList<String>();
    //Path para grabar los archivos en el directorio de trabajo
    String path=thepath;
    SAXBuilder builder = new SAXBuilder();
    //Leer el archivo descargado de OSM
    File xmlFile = new File(path.substring(0, path.length() -
8)+".osm");
    try {
        Document document = (Document) builder.build(xmlFile);
        Element rootNode = document.getRootElement();
        //Hacer una lista con todos los elementos que tienen way
        List list = rootNode.getChildren("way");
        for (int i = 0; i < list.size(); i++) {
            Element node = (Element) list.get(i);
            //Hacer una lista con todos los elementos que tienen
tag
            List listtag = node.getChildren("tag");
            //numero de elementos que tienen tag
            int sizeoflist=listtag.size();
            int theelement=i;
            info=information(sizeoflist,listtag,theelement);
            int valor;//tiene o no tiene nombre
            int pos;//posicion de etiqueta nombre
            int notway;//puede ser ocupada por vehiculo
            valor=info.retunrhavename();
            pos=info.retunrpos();
            notway=info.retunrnotway();

```

```

        if(pos!=-1)
        {
            Element tag = (Element) listtag.get(pos);
            if(valor==1)//Si tiene nombre
            {
                String nameofstreet;

nameofstreet=tag.getAttribute("v").toString().substring(15);
                nameofstreet=nameofstreet.substring(0,
nameofstreet.length() - 2);
                String idofstreet;

idofstreet=node.getAttribute("id").toString().substring(16);
                idofstreet=idofstreet.substring(0,
idofstreet.length() - 2);
                listOsmWays.add(nameofstreet+"/"+idofstreet);

cmbRoutes.addItem(nameofstreet+"/"+idofstreet);//+"-"+lareferencia);
            }
            else if(valor==0&&notway==0)//No tiene nombre pero si
es para simulacion
            {
                String idofstreet;

idofstreet=node.getAttribute("id").toString().substring(16);
                idofstreet=idofstreet.substring(0,
idofstreet.length() - 2);
                cmbRoutes.addItem("No
information"+"/"+idofstreet);// + "-" +lareferencia);
                listOsmWays.add("No information"+"/"+idofstreet);
            }
        }
    }
    catch (IOException io) {
        System.out.println(io.getMessage());
    } catch (JDOMException jdomex) {
        System.out.println(jdomex.getMessage());
    }
}

public void ejecutarComandoPolyConvertPersonal(int numberVehicles)
throws IOException, InterruptedException, SUMOException{
    AppConfiguration conf = new AppConfiguration();
    conf.load();
    String path=thepath;
    pathNetXML = path.substring(0, path.length()-
8).concat(Extensions.OSM_2017.getExtension());
    File fichero = new File("typemap.xml");

System.out.println("c4r.items.network.OSMMap.ejecutarComandoPolyConvert()
"+fichero.getAbsolutePath());
    String pathFileTypeMap = path.substring(0,
path.lastIndexOf('\\')+1).concat("typemap.xml");
    String nameFilePoly = path.substring(0, path.length() - 8) +
Extensions.POLY.getExtension();
    String nameFileRoute = path.substring(0, path.length() - 8) +
Extensions.ROUTE.getExtension();
}

```

```

        final String cmd = " --net-file " + path.replace(".c4r.xml",
".net.xml") + " --osm-files "+pathNetXML+ " --type-file "+
pathFileTypeMap + " -o "+nameFilePoly; //SUMO netconvert v0.31.0
        final String cmdPython1 = " python " +
conf.getSumoPath().substring(0, conf.getSumoPath().length()-3) +
"\\tools\\randomTrips.py "+
                                " -n " + path + " -e "+numberVehicles+
" -l";
        final String cmdPython2 = " python " +
conf.getSumoPath().substring(0, conf.getSumoPath().length()-3) +
"\\tools\\randomTrips.py "+
                                " -n " + path + " -r " +nameFileRoute+
" -e "+numberVehicles+ " -l";
        if (ejecutarComando(cmd, "\\polyconvert.exe")) {
            ejecutarComando(cmdPython1, null);
            ejecutarComando(cmdPython2, null);
        }
    }

    public boolean parseToSUMONet() throws IOException,
InterruptedException, SUMOException {
        String path=thepath;
        //SUMO netconvert v0.31.0
        final String cmd = " --osm-files " + path + " -o " + pathNetXML;
        String execution = "";
        if (SOValidator.isWindows()) {
            AppConfiguraton conf = new AppConfiguraton();
            conf.load();
            execution = conf.getSumoPath() + "\\netconvert.exe" + cmd;
        } else if (SOValidator.isUnix()) {
            execution ="netconvert" + cmd;
        }
        System.out.println ("Esto se ejecuta:::::" + execution);
        Process p = Runtime.getRuntime().exec(execution);

        /*Cleans the output and error buffers to avoid hang the
process.*/
        BufferCleaner bcOut = new BufferCleaner(p.getInputStream());
        BufferCleaner bcError = new BufferCleaner(p.getErrorStream());

        bcOut.start();
        bcError.start();
        int outValue = p.waitFor();
        if (outValue != 0) {
            /*Error occurred*/
            String nameLog = path.substring(0, path.length() - 8) +
Extensions.LOG.getExtension();
            PlainTextIO.save(nameLog, bcError.getOutput());
            throw new SUMOException("Impossible convert the map to SUMO
format.\n"
                                + "Further information in " + nameLog);
        }
        return outValue == 0;
    }

    private boolean ejecutarComando(String cmd, String instruccion)
throws IOException, InterruptedException, SUMOException{
        String path=thepath.replace("c4r.xml", "net.xml");
        String execution = "";

```

```

if (SOValidator.isWindows()) {
    AppConfiguration conf = new AppConfiguration();
    conf.load();
    if (instruccion == null) {
        execution = cmd.replace("c4r.xml", "net.xml");
    } else {
        String valor;
        valor=conf.getSumoPath();
        execution = conf.getSumoPath() + instruccion + cmd;
    }

} else if (SOValidator.isUnix()) {
    //execution = "sumo-netconvert" + cmd; //SUMO v0.11.1
    if (instruccion == null) {
        execution = cmd;
    } else {
        execution =instruccion.substring(0, instruccion.length()-
4) + cmd; //SUMO v0.12.0
    }
}

Process p = Runtime.getRuntime().exec(execution);

/*Cleans the output and error buffers to avoid hang the
process.*/
BufferCleaner bcOut = new BufferCleaner(p.getInputStream());
BufferCleaner bcError = new BufferCleaner(p.getErrorStream());

bcOut.start();
bcError.start();

int outValue = p.waitFor();

if (outValue != 0) {
    /*Error occurred*/
    String nameLog = path.substring(0, path.length() - 8) +
Extensions.LOG.getExtension();

    PlainTextIO.save(nameLog, bcError.getOutputStream());
    throw new SUMOException("Impossible convert the map to SUMO
format.\n"
        + "Further information in " + nameLog);
}
return outValue == 0;
}

public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel
setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay
with the default look and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {

```

```

        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels ()) {
            if ("Nimbus".equals(info.getName ())) {
javax.swing.UIManager.setLookAndFeel (info.getClassName ());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(JFrameRoutes.class.getName ()).log (java
.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(JFrameRoutes.class.getName ()).log (java
.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(JFrameRoutes.class.getName ()).log (java
.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(JFrameRoutes.class.getName ()).log (java
.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable () {
        public void run () {
            new JFrameRoutes ("").setVisible (true);

        }

    });
}

// Variables declaration - do not modify
private javax.swing.JButton btnAccept;
private javax.swing.JButton btnAdd;
private javax.swing.JButton btnAddCar;
private javax.swing.JButton btnCreateRou;
private javax.swing.JButton btnCreateRoute;
private javax.swing.JButton btnRemove;
private javax.swing.JButton btnRemoveCar;
private javax.swing.JButton btnRemoveRoute;
private javax.swing.JButton btnReset;
private javax.swing.JButton btnSimulation;
private java.awt.Choice cmbColor;
private java.awt.Choice cmbDepart;
private java.awt.Choice cmbFollowModel;
private java.awt.Choice cmbRoutes;
private java.awt.Choice cmbVehicleType;
private javax.swing.JInternalFrame jInternalFrame1;
private java.awt.Label label1;
private java.awt.Label label10;
private java.awt.Label label11;
private java.awt.Label label12;
private java.awt.Label label13;
private java.awt.Label label2;

```

```
private java.awt.Label label3;  
private java.awt.Label label4;  
private java.awt.Label label5;  
private java.awt.Label label6;  
private java.awt.Label label7;  
private java.awt.Label label8;  
private java.awt.Label label9;  
private java.awt.List lstRoutes;  
private java.awt.List lstRoutesCreated;  
private java.awt.List lstRoutesFinal;  
private java.awt.List lstRoutesName;  
private java.awt.List lstVehicle;  
private java.awt.TextField txtAccel;  
private java.awt.TextField txtDepart;  
private java.awt.TextField txtID;  
private java.awt.TextField txtSpeed;  
private java.awt.TextField txtTotalCars;  
private java.awt.TextField txtnCars;  
// End of variables declaration  
}
```


ANEXO II. ARCHIVO DE RUTAS GENERADO POR SOFTWARE MEJORADO

En esta sección se muestra el contenido del archivo de rutas generado mediante el programa C4R mejorado, en donde se puede encontrar que las vías que son utilizadas por los vehículos son las apropiadas y las exclusivas para otros tipos de transporte o de trenes (en el caso del ejemplo) no son ocupadas.

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- generated on 05/28/18 22:47:38 by SUMO duarouter Version 0.31.0
<?xml version="1.0" encoding="UTF-8"?>

<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/duarouterConfigurat
ion.xsd">

  <input>
    <net-file
value="C:\Users\SantiagoOate\Documents\NewYork\NewYork.net.xml"/>
    <route-files value="trips.trips.xml"/>
  </input>

  <output>
    <output-file
value="C:\Users\SantiagoOate\Documents\NewYork\NewYork.rou.xml"/>
  </output>

  <processing>
    <ignore-errors value="true"/>
  </processing>

  <time>
    <begin value="0"/>
    <end value="100.0"/>
  </time>

  <report>
    <no-warnings value="true"/>
    <no-step-log value="true"/>
  </report>

</configuration>
-->

<routes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/routes file.xsd">
  <vehicle id="1" depart="1.00">
    <route edges="222299238#0 222299238#1"/>
  </vehicle>
  <vehicle id="2" depart="2.00">
    <route edges="483707770 -275016228#1 196116979 196116978#0
196116978#1 196116978#2 587038049#0 587038049#1 587038049#2 388318981#0
```

```

388318981#1 388318981#2 222299275#0 222299275#1 541345307#2
541345307#3"/>
</vehicle>
<vehicle id="3" depart="3.00">
  <route edges="413050335#0 46171884#1 46171884#2 5669882#0
5669882#1 5669882#2 587038050#0 587038050#1 587038049#0 587038049#1
587038049#2 46116329#0 46116329#1 222299277"/>
</vehicle>
<vehicle id="4" depart="4.00">
  <route edges="275016228#0 196116979 196116978#0 25026344
222299272"/>
</vehicle>
<vehicle id="5" depart="5.00">
  <route edges="413050335#2 588168462 196116978#0 196116978#1
196116978#2 587038049#0 587038049#1 587038049#2 388318981#0 388318981#1
388318981#2 46116326#0 46116326#1 46116326#2 46116326#3 455501322
5671587#0 5671587#1"/>
</vehicle>
<vehicle id="7" depart="7.00">
  <route edges="541345307#1 222299275#2 222299275#3 222299275#4
222299275#5 222299275#6 222299275#7 222299275#8 222299275#9"/>
</vehicle>
<vehicle id="8" depart="8.00">
  <route edges="541345307#1 541345307#2 541345307#3 541345307#4
5672377#2 5672377#3 5672377#4 222299277 587038050#0"/>
</vehicle>
<vehicle id="9" depart="9.00">
  <route edges="496982627#0 496982627#1 420810619 196116979
196116978#0 25026344 222299254 483707770"/>
</vehicle>
<vehicle id="10" depart="10.00">
  <route edges="455501322 455501315 413056164 420888149 194923759#0
194923759#1 388318981#0 388318981#1 388318981#2 222299275#0 222299275#1
222299275#2 5673211"/>
</vehicle>
<vehicle id="11" depart="11.00">
  <route edges="-587038050#0 587038051 222299254 588168462
196116978#0 25026344"/>
</vehicle>
<vehicle id="12" depart="12.00">
  <route edges="194923759#0 194923759#1 388318981#0 388318981#1
388318981#2 46116326#0 46116326#1 46116326#2 46116326#3 455501322
5671587#0 5671587#1"/>
</vehicle>
<vehicle id="13" depart="13.00">
  <route edges="5673530#0 5673530#1 222299275#0 222299275#1
541345307#2 541345307#3 541345307#4 5672377#2 5672377#3 5672377#4
222299277 587038050#0 587038050#1 587038049#0 587038049#1"/>
</vehicle>
<vehicle id="14" depart="14.00">
  <route edges="222299275#2 222299275#3 222299275#4 222299275#5"/>
</vehicle>
<vehicle id="15" depart="15.00">
  <route edges="46160365#2 46160153#0 46160153#1 46160153#2
455501315 413056164 420888149 194923759#0 194923759#1 388318981#0
388318981#1 388318981#2 222299275#0 222299275#1 222299275#2 222299275#3
222299275#4 222299275#5 222299275#6"/>
</vehicle>
<vehicle id="16" depart="16.00">

```

```

    <route edges="-587038050#0 587038050#0 587038050#1 587038049#0
587038049#1 587038049#2 388318981#0 388318981#1 388318981#2 222299275#0
222299275#1 222299275#2 5673211"/>
  </vehicle>
  <vehicle id="17" depart="17.00">
    <route edges="140391522#3 46160365#2 46160153#0 46160153#1
46160153#2 455501315 413056164 420888149 194923759#0 194923759#1
46116329#0 46116329#1 222299277 587038051 222299254 483707770 -
275016228#1"/>
  </vehicle>
  <vehicle id="18" depart="18.00">
    <route edges="46116329#1 222299238#0 222299238#1 222299238#2"/>
  </vehicle>
  <vehicle id="19" depart="19.00">
    <route edges="388318981#1 388318981#2 46116326#0 46116326#1
46116326#2 46116326#3 455501322 5671587#0 5671587#1 5671587#2 494221658#0
494221658#1"/>
  </vehicle>
  <vehicle id="21" depart="21.00">
    <route edges="455501328#1 455501328#2 413056164 420888149
194923759#0 194923759#1 46116329#0 46116329#1 222299277 587038051
222299272 46171879#0"/>
  </vehicle>
  <vehicle id="23" depart="23.00">
    <route edges="588168462 196116978#0 25026344 222299272
46171880"/>
  </vehicle>
  <vehicle id="24" depart="24.00">
    <route edges="-587038050#0 587038051 222299272 46171879#0
46171879#1 222299238#1 222299238#2 222299238#3"/>
  </vehicle>
  <vehicle id="25" depart="25.00">
    <route edges="222299275#6"/>
  </vehicle>
  <vehicle id="26" depart="26.00">
    <route edges="275016228#0 196116979 196116978#0 25026344"/>
  </vehicle>
  <vehicle id="27" depart="27.00">
    <route edges="541345307#0 541345307#1 541345307#2 541345307#3
541345307#4 5672377#2 5672377#3 5672377#4 222299277 587038051 222299272
46171879#0"/>
  </vehicle>
  <vehicle id="28" depart="28.00">
    <route edges="587038050#0 587038050#1 587038049#0 587038049#1
587038049#2 46116329#0 46116329#1 222299238#0 222299238#1 -413045905"/>
  </vehicle>
  <vehicle id="32" depart="32.00">
    <route edges="46160365#0 541345307#0 541345307#1 222299275#2
222299275#3 222299275#4 222299275#5 222299275#6 222299275#7 222299275#8
222299275#9 222299275#10"/>
  </vehicle>
  <vehicle id="34" depart="34.00">
    <route edges="46160365#2 46160153#0 46160153#1 46160153#2
455501315 413056164 420888149 194923759#0 194923759#1 46116329#0
46116329#1 222299277 587038051 222299254"/>
  </vehicle>
  <vehicle id="35" depart="35.00">
    <route edges="275016228#0 196116979 196116978#0 196116978#1
196116978#2 587038049#0 587038049#1 587038049#2 388318981#0 388318981#1
388318981#2 222299275#0 222299275#1 222299275#2 5673211"/>

```

```

</vehicle>
<vehicle id="37" depart="37.00">
  <route edges="275016228#0 196116979 196116978#0 196116978#1
196116978#2 587038049#0 587038049#1 587038049#2 388318981#0 388318981#1
388318981#2 140391522#0 140391522#1 140391522#2 140391522#3
140391522#4"/>
</vehicle>
<vehicle id="38" depart="38.00">
  <route edges="413050335#0 46171884#1 46171884#2 5669882#0
5669882#1 5669882#2 587038050#0 587038050#1 587038049#0 587038049#1
587038049#2 388318981#0 388318981#1 388318981#2 222299275#0 222299275#1
541345307#2 541345307#3 541345307#4 5672377#2 5672377#3 5673530#0"/>
</vehicle>
<vehicle id="41" depart="41.00">
  <route edges="222299275#2 5673211 5672377#1 5672377#2 5672377#3
5673530#0 5673530#1 140391522#0 140391522#1 140391522#2 140391522#3
46160365#2 46160153#0 46160153#1"/>
</vehicle>
<vehicle id="42" depart="42.00">
  <route edges="587038051 222299254 483707770 -275016228#1 -
275016228#0"/>
</vehicle>
<vehicle id="45" depart="45.00">
  <route edges="413050335#1 413050335#2 483707770 -275016228#1 -
275016228#0"/>
</vehicle>
<vehicle id="46" depart="46.00">
  <route edges="46171884#1 46171884#2 5669882#0 5669882#1 5669882#2
587038051 222299254 483707770 -275016228#1 275016228#1"/>
</vehicle>
<vehicle id="47" depart="47.00">
  <route edges="222299275#2 5673211 5672377#1 5672377#2 5672377#3
5672377#4 222299277 587038051 222299272 46171879#0 5669882#1"/>
</vehicle>
<vehicle id="48" depart="48.00">
  <route edges="46160365#1 46160365#2 46160153#0 46160153#1
46160153#2 455501315 413056164 420888149 194923759#0 194923759#1
388318981#0 388318981#1 388318981#2 222299275#0 222299275#1 222299275#2
222299275#3"/>
</vehicle>
<vehicle id="51" depart="51.00">
  <route edges="587038050#0 587038050#1 587038049#0 587038049#1
587038049#2 46116329#0 46116329#1 222299238#0 222299238#1 222299238#2"/>
</vehicle>
<vehicle id="52" depart="52.00">
  <route edges="5672377#0 5672377#1 5672377#2 5672377#3 5673530#0
5673530#1 222299275#0 222299275#1 222299275#2 222299275#3 222299275#4
222299275#5 222299275#6 222299275#7 222299275#8 222299275#9 222299275#10
222299275#11"/>
</vehicle>
<vehicle id="53" depart="53.00">
  <route edges="46160083 455501322 455501315 413056164 420888149
194923759#0 194923759#1 46116329#0 46116329#1 222299238#0 222299238#1
222299238#2"/>
</vehicle>
<vehicle id="54" depart="54.00">
  <route edges="455501328#1 455501328#2 413056164 420888149
194923759#0 194923759#1 46116329#0 46116329#1 222299277 587038051
222299254 483707770 -275016228#1 275016228#1"/>
</vehicle>

```

```

<vehicle id="55" depart="55.00">
  <route edges="5672377#1 5672377#2 5672377#3 5673530#0 5673530#1
222299275#0 222299275#1 222299275#2 222299275#3 222299275#4 222299275#5
222299275#6 222299275#7 222299275#8 222299275#9 222299275#10
222299275#11"/>
</vehicle>
<vehicle id="56" depart="56.00">
  <route edges="5673211 5672377#1 5672377#2 5672377#3 5673530#0
5673530#1 140391522#0 140391522#1 140391522#2 140391522#3"/>
</vehicle>
<vehicle id="57" depart="57.00">
  <route edges="46171879#1"/>
</vehicle>
<vehicle id="58" depart="58.00">
  <route edges="46160365#2 46160153#0 46160153#1 46160153#2
455501315 413056164 420888149 194923759#0 194923759#1 46116329#0
46116329#1 222299277 587038051 222299254 483707770 -275016228#1 -
275016228#0"/>
</vehicle>
<vehicle id="59" depart="59.00">
  <route edges="455501328#1 455501328#2 413056164 420888149
194923759#0 194923759#1 46116329#0 46116329#1 222299277 587038051
222299254 483707770 -275016228#1 -275016228#0"/>
</vehicle>
<vehicle id="60" depart="60.00">
  <route edges="413050335#0 46171884#1 46171884#2 5669882#0
5669882#1 5669882#2 587038050#0 587038050#1 587038049#0 587038049#1
587038049#2 388318981#0 388318981#1 388318981#2 46116326#0 46116326#1
46116326#2 46116326#3 455501322 5671587#0 5671587#1 5671587#2 25564540"/>
</vehicle>
<vehicle id="61" depart="61.00">
  <route edges="587038049#1 587038049#2 388318981#0 388318981#1
388318981#2 46116326#0 46116326#1 46116326#2 46116326#3 455501322
5671587#0 5671587#1 5671587#2 25564540"/>
</vehicle>
<vehicle id="62" depart="62.00">
  <route edges="275016228#0 275016228#1"/>
</vehicle>
<vehicle id="63" depart="63.00">
  <route edges="5672377#4 222299277 587038050#0 587038050#1
587038049#0 587038049#1 587038049#2 388318981#0 388318981#1 388318981#2
140391522#0 140391522#1 140391522#2 140391522#3 140391522#4"/>
</vehicle>
<vehicle id="66" depart="66.00">
  <route edges="5673038 46160153#0 46160153#1 46160153#2 455501315
413056164 420888149 194923759#0 194923759#1 46116329#0 46116329#1
222299277 587038051 222299254 588168462 196116978#0"/>
</vehicle>
<vehicle id="67" depart="67.00">
  <route edges="587038050#0 -587038050#0 587038051 222299272
46171879#0 5669882#1 5669882#2"/>
</vehicle>
<vehicle id="68" depart="68.00">
  <route edges="5669882#1 5669882#2 587038050#0 587038050#1
587038049#0 587038049#1 587038049#2 388318981#0 388318981#1 388318981#2
46116326#0 46116326#1 46116326#2 46116326#3 455501322"/>
</vehicle>
<vehicle id="69" depart="69.00">
  <route edges="-275016228#1 196116979 196116978#0 196116978#1
196116978#2 587038049#0 587038049#1 587038049#2 388318981#0 388318981#1

```

```

388318981#2 140391522#0 140391522#1 140391522#2 140391522#3
140391522#4"/>
</vehicle>
<vehicle id="70" depart="70.00">
  <route edges="413050335#1 413050335#2 588168462"/>
</vehicle>
<vehicle id="71" depart="71.00">
  <route edges="46160153#0 46160153#1 46160153#2 455501315
413056164 420888149 194923759#0 194923759#1 388318981#0 388318981#1
388318981#2 222299275#0 222299275#1 222299275#2 5673211"/>
</vehicle>
<vehicle id="73" depart="73.00">
  <route edges="388318981#0 388318981#1 388318981#2 222299275#0
222299275#1 222299275#2 222299275#3 222299275#4 222299275#5 222299275#6
222299275#7 222299275#8 222299275#9 222299275#10 222299275#11"/>
</vehicle>
<vehicle id="77" depart="77.00">
  <route edges="5669882#0 5669882#1 5669882#2 587038050#0
587038050#1 587038049#0 587038049#1 587038049#2 388318981#0 388318981#1
388318981#2 46116326#0 46116326#1 46116326#2 46116326#3 455501322
5671587#0 5671587#1 5671587#2 520532096"/>
</vehicle>
<vehicle id="78" depart="78.00">
  <route edges="455501328#0 455501328#1"/>
</vehicle>
<vehicle id="79" depart="79.00">
  <route edges="194923759#0 194923759#1 46116329#0 46116329#1"/>
</vehicle>
<vehicle id="83" depart="83.00">
  <route edges="46160365#0 46160365#1 46160365#2 46160153#0
46160153#1 46160153#2 455501315 413056164 420888149 194923759#0
194923759#1 46116329#0 46116329#1"/>
</vehicle>
<vehicle id="84" depart="84.00">
  <route edges="-587038050#0 587038050#0 587038050#1 587038049#0
587038049#1 587038049#2 388318981#0 388318981#1 388318981#2 222299275#0
222299275#1 222299275#2 5673211"/>
</vehicle>
<vehicle id="85" depart="85.00">
  <route edges="413050335#2 483707770 -275016228#1"/>
</vehicle>
<vehicle id="86" depart="86.00">
  <route edges="5673530#0 5673530#1 46116326#0 46116326#1
46116326#2 46116326#3 455501322"/>
</vehicle>
<vehicle id="87" depart="87.00">
  <route edges="388318981#1 388318981#2 222299275#0 222299275#1
541345307#2 541345307#3 541345307#4 5672377#2 5672377#3 5672377#4
222299238#0"/>
</vehicle>
<vehicle id="88" depart="88.00">
  <route edges="222299272 46171879#0 5669882#1 5669882#2
587038050#0 587038050#1 587038049#0 587038049#1 587038049#2 388318981#0
388318981#1 388318981#2 222299275#0 222299275#1 541345307#2 541345307#3
541345307#4 5672377#2 5672377#3 5673530#0"/>
</vehicle>
<vehicle id="89" depart="89.00">
  <route edges="588168462 196116978#0 25026344 222299272 46171880
46171884#2 46171884#3"/>
</vehicle>

```

```

    <vehicle id="90" depart="90.00">
      <route edges="5673038 46160153#0 46160153#1 46160153#2 455501315
413056164 420888149 194923759#0 194923759#1 46116329#0 46116329#1
222299277 587038050#0"/>
    </vehicle>
    <vehicle id="91" depart="91.00">
      <route edges="413050335#2 483707770 -275016228#1 -275016228#0"/>
    </vehicle>
    <vehicle id="92" depart="92.00">
      <route edges="5672377#2 5672377#3 5672377#4 222299277 587038050#0
-587038050#0"/>
    </vehicle>
    <vehicle id="94" depart="94.00">
      <route edges="25026344 222299254 588168462"/>
    </vehicle>
    <vehicle id="95" depart="95.00">
      <route edges="46160365#0 541345307#0 541345307#1 541345307#2
541345307#3 541345307#4 5672377#2 5672377#3 5672377#4 222299277 587038051
222299254 588168462"/>
    </vehicle>
    <vehicle id="96" depart="96.00">
      <route edges="496982627#1 420810619 196116979 196116978#0
196116978#1 196116978#2 587038049#0 587038049#1 587038049#2 388318981#0
388318981#1"/>
    </vehicle>
    <vehicle id="97" depart="97.00">
      <route edges="25026344 222299272 46171879#0 5669882#1 5669882#2
587038051"/>
    </vehicle>
    <vehicle id="98" depart="98.00">
      <route edges="588168462 196116978#0 196116978#1 196116978#2
587038049#0 587038049#1 587038049#2 388318981#0 388318981#1 388318981#2
222299275#0 222299275#1 222299275#2 5673211 5672377#1"/>
    </vehicle>
    <vehicle id="99" depart="99.00">
      <route edges="46171880 46171884#2 5669882#0 5669882#1 5669882#2
587038050#0 587038050#1 587038049#0 587038049#1 587038049#2 388318981#0
388318981#1 388318981#2 46116326#0 46116326#1 46116326#2 46116326#3
455501322 5671587#0 5671587#1 5671587#2 25564540"/>
    </vehicle>
  </routes>

```

ANEXO III. ARCHIVO DE RUTAS GENERADO POR SOFTWARE C4R EN VERSION ORIGINAL

En esta sección se muestra el contenido del archivo de rutas generado mediante el programa C4R original, en donde se puede encontrar que las vías que son utilizadas por los vehículos no son las apropiadas ya que las exclusivas de transporte o de trenes si esta siendo ocupadas.

```
<routes>
<routeDistribution id="undefined_0">
<route edges="-44225576#1 -44225576#0 5673530#0 5673530#1 140391522#0
140391522#1 -496162968#2" id="undefined_0_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_1">
<route edges="496162983#0 388318981#2 140391522#0 140391522#1 -
496162968#2 496162985" id="undefined_1_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_2">
<route edges="388318981#0 -496162987#0 -496162995#1 -496162995#0 -
496162981#2 -496162981#1" id="undefined_2_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_3">
<route edges="194753576 496162965#2 496162981#0 496162981#1 496162981#2
496162995#0 496162995#1 496162987#0 496162987#1" id="undefined_3_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_4">
<route edges="277649037 156061426#1 533819094#0 496162981#1 496162981#2
496162995#0 496162995#1 496162954#0 -496162984#2 46116326#1"
id="undefined_4_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_5">
<route edges="533819094#1" id="undefined_5_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_6">
<route edges="46116326#0 496162984#2 -496162954#0 -496162995#1 -
496162995#0 -496162981#2 -496162981#1 -533819094#0 -156061426#1"
id="undefined_6_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_7">
<route edges="587038049#1 587038049#2 388318981#0 388318981#1 388318981#2
222299275#0 222299275#1 541345307#0 157005691#0+541345307#1 541345307#2
5672377#1 -515665872#1" id="undefined_7_0" probability="1.0"/>
</routeDistribution>
```



```

<routeDistribution id="undefined_8">
<route edges="157005691#0+541345307#1" id="undefined_8_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_9">
<route edges="533819094#0 496162981#1 496162981#2 496162995#0"
id="undefined_9_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_10">
<route edges="-413045905 462421592 -44225576#2 -44225576#1 -44225576#0
5673530#0 -496162969#0" id="undefined_10_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_11">
<route edges="-496162995#1 -496162995#0 533819086 -481331159 -533819083#1
196116978#2" id="undefined_11_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_12">
<route edges="275016228 196116979 196116978#0 196116978#1"
id="undefined_12_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_13">
<route edges="447790221" id="undefined_13_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_14">
<route edges="-44225576#1 -44225576#0 5672377#3 222299277 587038051
222299254 588168462" id="undefined_14_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_15">
<route edges="587038049#1 -496162961#0 -496162981#2 -496162981#1 -
496162981#0 -496162965#2 -496162965#1 -496162965#0" id="undefined_15_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_16">
<route edges="-496162968#0 -496162954#1 -496162954#0 -496162995#1
194923759#1 46116329#0" id="undefined_16_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_17">
<route edges="-156061426#0 -156061426#4 -156060805 -533819083#1
196116978#2 -587038050#1" id="undefined_17_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_18">
<route edges="-44225576#2 -44225576#1 -44225576#0 5672377#3 222299277
587038050#0 -533819091#0 533819097 -496162961#1 -496162961#0 -
496162981#2" id="undefined_18_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_19">
<route edges="-44225576#1 -44225576#0 5672377#3 222299277 587038050#0 -
533819091#0 -533819089#1 587038049#1 -496162961#0 -496162981#2"
id="undefined_19_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_20">
<route edges="496162987#1" id="undefined_20_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_21">
<route edges="-157005691#1" id="undefined_21_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_22">
<route edges="515665872#1 5672377#2 5673530#0 5673530#1 46116326#0
46116326#1 46116326#2 455501322 5671587#0 -496162994#1"
id="undefined_22_0" probability="1.0"/>

```

```

</routeDistribution>
<routeDistribution id="undefined_23">
<route edges="-496162994#0 -533819094#1 496162981#1 496162981#2
496162961#0 496162961#1" id="undefined_23_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_24">
<route edges="533819091#1 533819083#0 533819083#1 156060805 -156061426#3
-156061426#2" id="undefined_24_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_25">
<route edges="515665872#0 515665872#1 5672377#2 5673530#0 5673530#1"
id="undefined_25_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_26">
<route edges="-496162954#0 -496162995#1 -496162995#0 -496162981#2 -
496162981#1 -533819094#0 -156061426#1" id="undefined_26_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_27">
<route edges="157005691#0+541345307#1 541345307#2 5672377#1 5672377#2
5673530#0 5673530#1 46116326#0 46116326#1 46116326#2 455501322 5671587#0"
id="undefined_27_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_28">
<route edges="5672459 -44225576#1 -44225576#0 5673530#0 5673530#1
46116326#0 46116326#1 46116326#2 455501322" id="undefined_28_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_29">
<route edges="-496162994#2 -496162994#1 -496162994#0 -533819094#1
496162981#1 496162981#2 533819086 -481331159 -533819083#1"
id="undefined_29_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_30">
<route edges="-496162981#2 -496162981#1 -496162981#0 -496162965#2 -
194753576 -186050568" id="undefined_30_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_31">
<route edges="-533819086" id="undefined_31_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_32">
<route edges="196116978#1 196116978#2 587038049#0 587038049#1 587038049#2
388318981#0 388318981#1 388318981#2 140391522#0 -496162984#0"
id="undefined_32_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_33">
<route edges="-482029239" id="undefined_33_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_34">
<route edges="391062195 482026799 196116978#1 -533819083#0"
id="undefined_34_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_35">
<route edges="5672377#1 5672377#2 44225576#0" id="undefined_35_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_36">
<route edges="533819083#0 533819083#1 481331159 -533819086 -496162981#2 -
496162981#1 -496162981#0 -496162965#2 -496162965#1 -496162965#0"
id="undefined_36_0" probability="1.0"/>

```

```

</routeDistribution>
<routeDistribution id="undefined_37">
<route edges="496162981#0 -533819094#0 -156061426#1 -277649037 -25026345
391062195 482026799 25026344 222299254 588168462" id="undefined_37_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_38">
<route edges="-496162968#2 -496162968#1 -496162968#0 -496162954#1 -
496162954#0 -496162995#1 194923759#1 46116329#0 46116329#1 222299277
587038051 222299272 46171879#0 5669882#1" id="undefined_38_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_39">
<route edges="494221658#0 -496162971 -496162966 -496162992#1 -496162992#0
-496162985 496162968#2 496162968#3 -496162982 -496162972#1 222299275#1
541345307#0 157005691#0+541345307#1 541345307#2" id="undefined_39_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_40">
<route edges="388318981#0 388318981#1 496162983#1 496162954#1 496162968#0
46116326#2 455501322 5671587#0 5671587#1 5671587#2 25564540"
id="undefined_40_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_41">
<route edges="222299272 46171879#0 46171879#1 5672459 -44225576#1 -
44225576#0 5673530#0 496162969#1 496162972#0 222299275#1 541345307#0
157005691#0+541345307#1 157005691#1" id="undefined_41_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_42">
<route edges="-496162994#2" id="undefined_42_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_43">
<route edges="496162965#0 5671587#2" id="undefined_43_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_44">
<route edges="277649037 156061426#1 533819094#0 496162981#1 496162981#2
496162995#0 496162995#1 496162954#0 -496162984#2" id="undefined_44_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_45">
<route edges="496162965#1 496162965#2 496162981#0 -533819094#0 -
156061426#1 -277649037 -25026345" id="undefined_45_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_46">
<route edges="496162954#0 -496162984#2" id="undefined_46_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_47">
<route edges="-496162954#1 -496162983#1" id="undefined_47_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_48">
<route edges="5672459 -44225576#1 -44225576#0 5673530#0 5673530#1
46116326#0 46116326#1 496162968#1" id="undefined_48_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_49">

```

```

<route edges="482026799 25026344 222299272 46171879#0 46171879#1
222299238#1" id="undefined_49_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_50">
<route edges="-482029239" id="undefined_50_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_51">
<route edges="496162992#1 496162966 496162965#0 496162965#1 496162965#2"
id="undefined_51_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_52">
<route edges="46116329#1 222299277 587038051 222299254 588168462
196116978#0" id="undefined_52_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_53">
<route edges="455501315 413056164 420888149 194923759#0 496162995#1
496162954#0 -496162983#1 -496162983#0 496162969#0 496162969#1"
id="undefined_53_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_54">
<route edges="-157005691#1 541345307#2 5672377#1 5672377#2 5672377#3
222299277 587038050#0 -533819091#0 -533819089#1 -533819089#0 -533819086"
id="undefined_54_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_55">
<route edges="-496162968#1 -496162968#0 -496162954#1 -496162954#0 -
496162995#1 -496162995#0 -496162981#2 -277644851#1" id="undefined_55_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_56">
<route edges="5673530#0 5673530#1 46116326#0 46116326#1 496162968#1"
id="undefined_56_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_57">
<route edges="496162968#0 46116326#2 455501322 5671587#0 5671587#1
496162965#1 496162965#2" id="undefined_57_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_58">
<route edges="156061426#0 -277649037 -25026345 391062195 482026799
25026344" id="undefined_58_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_59">
<route edges="277644851#1 -496162981#1 533819094#1 496162994#0
496162994#1 496162994#2 -496162992#1" id="undefined_59_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_60">
<route edges="420888149 194923759#0 496162995#1 496162954#0"
id="undefined_60_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_61">
<route edges="5672377#0" id="undefined_61_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_62">
<route edges="455501315 413056164 420888149 194923759#0 194923759#1
46116329#0 496162989#1" id="undefined_62_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_63">
<route edges="496162954#0 -496162984#2 -496162984#1 140391522#1"
id="undefined_63_0" probability="1.0"/>

```

```

</routeDistribution>
<routeDistribution id="undefined_64">
<route edges="-496162966 -496162992#1 -496162992#0 -496162985 496162968#2
496162968#3" id="undefined_64_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_65">
<route edges="-44225576#2 -44225576#1 -44225576#0 5672377#3 222299277
587038051 222299254 588168462 196116978#0" id="undefined_65_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_66">
<route edges="46171879#1 5672459 -44225576#1 -44225576#0 5673530#0"
id="undefined_66_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_67">
<route edges="-496162982 496162984#0 496162984#1 496162984#2 -496162954#0
-496162995#1 -496162995#0 533819086 -481331159 156060805"
id="undefined_67_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_68">
<route edges="-496162994#1 -496162994#0 -533819094#1 496162981#1"
id="undefined_68_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_69">
<route edges="5669882#2 587038050#0 -533819091#0 533819097 496162989#0
496162989#1 -496162956 496162969#0 5673530#1 140391522#0 140391522#1"
id="undefined_69_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_70">
<route edges="-496162983#0 496162956 -496162989#1" id="undefined_70_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_71">
<route edges="515665872#0 515665872#1 5672377#2 5672377#3 222299277
587038050#0 -533819091#0" id="undefined_71_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_72">
<route edges="-496162984#0 -496162972#1 222299275#1 541345307#0"
id="undefined_72_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_73">
<route edges="496162984#0 140391522#1" id="undefined_73_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_74">
<route edges="5672377#2 5673530#0 496162969#1" id="undefined_74_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_75">
<route edges="-496162989#1 46116329#1 222299277 587038050#0"
id="undefined_75_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_76">
<route edges="447790221" id="undefined_76_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_77">
<route edges="-496162972#0 -496162969#1 -496162969#0 496162956 -
496162989#1 46116329#1 222299277 587038051 222299272" id="undefined_77_0"
probability="1.0"/>
</routeDistribution>

```

```

<routeDistribution id="undefined_78">
<route edges="-496162965#2 -496162965#1 -496162965#0" id="undefined_78_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_79">
<route edges="462421592" id="undefined_79_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_80">
<route edges="-44225576#0 5672377#3" id="undefined_80_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_81">
<route edges="551182884 194753576 496162965#2 496162981#0 -533819094#0 -
156061426#1 -277649037 -25026345 391062195 482026799 25026344"
id="undefined_81_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_82">
<route edges="494221658#0 -496162971 496162965#0 496162965#1 496162965#2
496162981#0 496162981#1 496162981#2 496162961#0 496162961#1 496162989#0
46116329#1 222299238#0 5672459 -44225576#1" id="undefined_82_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_83">
<route edges="496162984#2 -496162954#0 -496162995#1 -496162995#0 -
496162981#2 -496162981#1 -533819094#0 -156061426#1 -277649037"
id="undefined_83_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_84">
<route edges="496162983#1 -496162954#0 -496162995#1 -496162995#0 -
496162981#2 -277644851#1 -277644851#0+496162988" id="undefined_84_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_85">
<route edges="515665871 515665872#1 5672377#2 5672377#3 222299238#0
222299238#1 -413045905" id="undefined_85_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_86">
<route edges="196116978#0 25026344 222299272 46171879#0 46171879#1
5672459 44225576#2" id="undefined_86_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_87">
<route edges="587038049#0 587038049#1 496162961#1" id="undefined_87_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_88">
<route edges="587038049#1" id="undefined_88_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_89">
<route edges="-44225576#0 5672377#3 222299277 587038050#0 587038050#1
587038049#0 587038049#1 -496162961#0 -496162981#2 -496162981#1 -
496162981#0" id="undefined_89_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_90">
<route edges="5672459 -44225576#1 -44225576#0 5673530#0 5673530#1
222299275#0 222299275#1 541345307#0" id="undefined_90_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_91">
<route edges="5673530#1 140391522#0 140391522#1 496162968#3"
id="undefined_91_0" probability="1.0"/>

```

```

</routeDistribution>
<routeDistribution id="undefined_92">
<route edges="-496162994#1 455501328#1 413056164 420888149 194923759#0
194923759#1 46116329#0" id="undefined_92_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_93">
<route edges="46171879#0 5669882#1 5669882#2 587038050#0 587038050#1
587038049#0 587038049#1" id="undefined_93_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_94">
<route edges="533819094#1 496162994#0" id="undefined_94_0"
probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_95">
<route edges="-44225576#1 -44225576#0 5673530#0 -496162969#0"
id="undefined_95_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_96">
<route edges="-482029239" id="undefined_96_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_97">
<route edges="447790221" id="undefined_97_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_98">
<route edges="-413045905 462421592 -44225576#2 -44225576#1 -44225576#0
5673530#0 -496162969#0 496162956 -496162987#1 -496162987#0"
id="undefined_98_0" probability="1.0"/>
</routeDistribution>
<routeDistribution id="undefined_99">
<route edges="196116978#0 196116978#1" id="undefined_99_0"
probability="1.0"/>
</routeDistribution>
<vehicle depart="0" id="random_undefined_0_0" route="undefined_0"/>
<vehicle depart="0" id="random_undefined_1_0" route="undefined_1"/>
<vehicle depart="0" id="random_undefined_2_0" route="undefined_2"/>
<vehicle depart="0" id="random_undefined_3_0" route="undefined_3"/>
<vehicle depart="0" id="random_undefined_4_0" route="undefined_4"/>
<vehicle depart="0" id="random_undefined_5_0" route="undefined_5"/>
<vehicle depart="0" id="random_undefined_6_0" route="undefined_6"/>
<vehicle depart="0" id="random_undefined_7_0" route="undefined_7"/>
<vehicle depart="0" id="random_undefined_8_0" route="undefined_8"/>
<vehicle depart="0" id="random_undefined_9_0" route="undefined_9"/>
<vehicle depart="0" id="random_undefined_10_0" route="undefined_10"/>
<vehicle depart="0" id="random_undefined_11_0" route="undefined_11"/>
<vehicle depart="0" id="random_undefined_12_0" route="undefined_12"/>
<vehicle depart="0" id="random_undefined_13_0" route="undefined_13"/>
<vehicle depart="0" id="random_undefined_14_0" route="undefined_14"/>
<vehicle depart="0" id="random_undefined_15_0" route="undefined_15"/>
<vehicle depart="0" id="random_undefined_16_0" route="undefined_16"/>
<vehicle depart="0" id="random_undefined_17_0" route="undefined_17"/>
<vehicle depart="0" id="random_undefined_18_0" route="undefined_18"/>
<vehicle depart="0" id="random_undefined_19_0" route="undefined_19"/>
<vehicle depart="0" id="random_undefined_20_0" route="undefined_20"/>
<vehicle depart="0" id="random_undefined_21_0" route="undefined_21"/>
<vehicle depart="0" id="random_undefined_22_0" route="undefined_22"/>
<vehicle depart="0" id="random_undefined_23_0" route="undefined_23"/>
<vehicle depart="0" id="random_undefined_24_0" route="undefined_24"/>
<vehicle depart="0" id="random_undefined_25_0" route="undefined_25"/>
<vehicle depart="0" id="random_undefined_26_0" route="undefined_26"/>
<vehicle depart="0" id="random_undefined_27_0" route="undefined_27"/>

```



```
<vehicle depart="0" id="random_undefined_88_0" route="undefined_88"/>
<vehicle depart="0" id="random_undefined_89_0" route="undefined_89"/>
<vehicle depart="0" id="random_undefined_90_0" route="undefined_90"/>
<vehicle depart="0" id="random_undefined_91_0" route="undefined_91"/>
<vehicle depart="0" id="random_undefined_92_0" route="undefined_92"/>
<vehicle depart="0" id="random_undefined_93_0" route="undefined_93"/>
<vehicle depart="0" id="random_undefined_94_0" route="undefined_94"/>
<vehicle depart="0" id="random_undefined_95_0" route="undefined_95"/>
<vehicle depart="0" id="random_undefined_96_0" route="undefined_96"/>
<vehicle depart="0" id="random_undefined_97_0" route="undefined_97"/>
<vehicle depart="0" id="random_undefined_98_0" route="undefined_98"/>
<vehicle depart="0" id="random_undefined_99_0" route="undefined_99"/>
</routes>
```

ORDEN DE EMPASTADO