

UNIVERSIDAD DE GRANADA

**DEPARTAMENTO DE LENGUAJES Y
SISTEMAS INFORMÁTICOS**

**Escuela Técnica Superior de Ingenierías
Informática y Telecomunicaciones**

INGENIERÍA INFORMÁTICA

PROYECTO FIN DE CARRERA

Traffic Flow Simulator

Juan Antonio Fajardo Serrano

Director del proyecto: Domingo Martín Perandrés

Curso 2014/2015

Índice

1	Introducción	3
2	Objetivos	5
2.1	Objetivo principal del proyecto	5
2.2	Red viaria	5
2.3	Señales de tráfico y peatones	5
2.4	Vehículos	5
2.5	Conductores	6
2.6	Entorno	6
2.7	Uso de redes viarias reales	6
3	Estado del arte	7
3.1	Simuladores de tráfico	7
3.2	Representación de redes viarias	9
3.3	Sistemas de partículas conductistas	10
3.4	Modelos de redes viarias reales a partir de imágenes	10
3.5	Herramientas gráficas	11
4	Propuesta	12
4.1	Objetivo principal del proyecto	12
4.2	Red viaria	12
4.3	Señales de tráfico y peatones	17
4.4	Vehículos	17
4.5	Conductores	17
4.6	Entorno	17
4.7	Uso de redes viarias reales	17
	Bibliografía	18

Capítulo 1

Introducción

La presente memoria documenta el desarrollo de un simulador de tráfico, denominado *Traffic Flow Simulator*, que tendrá como base una interfaz gráfica desde la que el usuario podrá introducir cambios en las variables involucradas en la simulación y ver los resultados que ocasionan.

La simulación está definida como una representación de cierta parte del mundo real conseguida mediante la construcción de un modelo de ordenador que va evolucionando a lo largo del tiempo [1]. Según *Simulation of traffic systems* [2], el desarrollo de simuladores de tráfico mediante ordenador comenzó en 1955 con la tesis de Daniel L. Gerlough en la Universidad de California. En la actualidad, gracias al poder de cálculo de los computadores, se han logrado grandes avances en este campo, como algunos simuladores a escala macroscópica.

Este proyecto nace del problema actual de tráfico en las grandes ciudades y la necesidad de mejorar el flujo de vehículos en puntos concretos de las zonas urbanas, que es actualmente un problema diario.



Traffic Flow Simulator tiene como objetivo principal ser una herramienta útil que facilite la visualización de la simulación de tráfico en base a un modelo lógico, de forma que permita evaluar de manera clara las diferentes alternativas a situaciones concretas, permitiendo al usuario recrear dichas situaciones mediante la interacción con la herramienta teniendo en cuenta puntos importantes como son la congestión de las vías o los comportamientos de conducción, de forma que se pueda mejorar el flujo de tráfico en las vías urbanas. El segundo objetivo de la herramienta es posibilitar el uso de mapas de carreteras reales dentro del simulador.

El flujo de tráfico de vehículos es un sistema complejo y difícil de modelar y medir, ya que en este se ven involucradas multitud de variables dependientes unas de otras, entre las que destacan los problemas de movilidad, los cambios de sentido, las rotondas, los tiempos de semaforización, las obras en la calzada, los accidentes de tráfico y otras variables sin contar con el factor humano y las condiciones atmosféricas.

Este simulador posibilitará realizar simulaciones de tráfico sobre distintas distribuciones de redes viarias, teniendo en cuenta los diferentes perfiles de conductores que podemos encontrarnos día a día en la carretera. Así mismo, el usuario podrá personalizar las distintas variables que influirán en la simulación de forma que pueda recrear los escenarios que desee.

La parte visual estará compuesta por un entorno pasivo que no afectará al tráfico y otro activo que afectará directamente al tráfico como son semáforos, peatones y diferentes señales.

Capítulo 2

Objetivos

En este capítulo se mostrarán los objetivos que han sido marcados para la realización de este proyecto.

2.1 Objetivo principal del proyecto

El objetivo principal de este proyecto es la obtención de una aplicación multiplataforma de código abierto que permita observar, de forma gráfica en tres dimensiones, el flujo de tráfico de vehículos en distintas configuraciones de red viaria, que el usuario podrá cambiar mediante la interacción con la aplicación.

2.2 Red viaria

2.3 Señales de tráfico y peatones

2.4 Vehículos

Los vehículos que intervendrán en la simulación serán de al menos dos tipos, coches, y camiones no articulados.

Los modelos de estos vehículos serán de poco nivel de detalle con el fin de poder incluir en la simulación tantos como se pueda sin que se vea afectado el rendimiento del sistema.

El usuario podrá cambiar el número de vehículos involucrados en la simulación mediante la interacción con la interfaz gráfica.

2.5 Conductores

Para simular el flujo de tráfico de una forma más realista, el sistema conductista contará con al menos tres tipos de conductores, aquellos que cumplen las normas de circulación siempre, aquellos que las cumplen normalmente y por último, aquellos que casi nunca las cumplen, en menor proporción respecto a los otros dos tipos.

2.6 Entorno

Para obtener una simulación con cierto nivel de detalle se realizará un entorno de las redes viarias con algunos edificios, árboles, señales, semáforos, rotondas, pasos de peatones y paradas de autobús.

Así mismo contará con un ciclo diario en el que se podrá cambiar la velocidad del paso del tiempo. Las distintas velocidades serán: 1x, 5x, 15x, 30x, 60x, 120x.

2.7 Uso de redes viarias reales

Capítulo 3

Estado del arte

En el presente capítulo se abordará el estado del arte de los distintos elementos de los que se hará uso en el desarrollo de este proyecto.

En el apartado *Simuladores de tráfico* se presentará la esencia del proyecto; en el apartado *Representación de redes viarias*, se tratará una manera de trabajar con los mapas de redes viarias de forma que podamos generar el entorno en el que se desenvolverán los vehículos en nuestro simulador; en el apartado *Sistemas de partículas conductistas* se presentará el modelo de interacción utilizado por los vehículos mencionados anteriormente; en el apartado *Modelos de redes viarias reales a partir de imágenes* abordaremos un método mediante el cual podremos extraer la información de las redes viarias a partir de imágenes; finalmente, en el apartado *Herramientas gráficas* veremos las herramientas que se vienen utilizando para la creación de simuladores de tráfico.

3.1 Simuladores de tráfico

En general, la simulación está definida como una representación de cierta parte del mundo real conseguida mediante la construcción de un modelo de ordenador que va evolucionando a lo largo del tiempo [1].

Un simulador de tráfico es un modelo de ordenador de alguna parte de los sistemas de transporte del mundo real, cuya principal finalidad es la de ayudar a planear, diseñar y operar mejor dichos sistemas de transporte.

Se estima que la simulación nació en 1777 con el planteamiento del problema conocido como *La aguja de Buffon*. Este es un problema clásico de probabilidad geométrica, de realización práctica y cuyo interés radica en que es un método difícil para ir aproximando el valor del número π a partir de sucesivos intentos. Fue planteado por el naturalista francés Buffon en 1733

y reproducido por él mismo ya resuelto en 1757. Se trata de lanzar una aguja sobre un papel en el que se han trazado rectas paralelas distanciadas entre sí de manera uniforme. Se puede demostrar que si la distancia entre las rectas es igual a la longitud de la aguja, la probabilidad de que la aguja cruce alguna de las líneas es $2/\pi$.

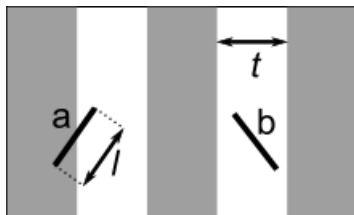


Figura 3.1: La aguja de Buffon

Los sistemas de simulación de transporte hicieron su aparición hace cuarenta años y la simulación por ordenador comenzó cuando D.L. Gerlough publicó su discurso: *Simulation of freeway traffic on a general-purpose discrete variable computer* (Simulación del tráfico de la autopista en un ordenador de propósito general de variables discretas.) en la Universidad de California, Los Ángeles, en 1955 [3].

Desde entonces, la simulación por ordenador se ha convertido en una herramienta ampliamente utilizada en la ingeniería del transporte con muchas aplicaciones que van desde la investigación científica hasta la planificación y el entrenamiento.

Actualmente, las líneas de investigación que se están siguiendo en este campo son mayoritariamente simulaciones microscópicas aunque también hay novedades muy interesantes en los modelos teóricos macroscópicos.

Como ejemplo de simulador de tráfico podemos nombrar *CORSIM*, el cual permite representar en dos dimensiones una red viaria y simular el flujo del tráfico a lo largo del tiempo indicando una serie de parámetros.

En el proyecto que nos ocupa, se va a elaborar un simulador de tráfico microscópico en tres dimensiones que permitirá analizar distintos tipos de configuraciones en una porción limitada de una red viaria.

3.2 Representación de redes viarias

Como se explica en el libro *Fundamentals of Traffic Simulation* [4], la representación de mapas o de las redes viarias dependerá del tipo de análisis que se quiera llevar a cabo sobre el sistema de transporte.

Inicialmente, la estructura más sencilla para representar lógicamente un mapa será un grafo dirigido, cuyos nodos serán las intersecciones y cuyos arcos serán las secciones viarias que conectan dichas intersecciones, indicando la dirección de las mismas.

Para completar este modelo, se caracterizan los arcos del grafo con atributos como, por ejemplo, *capacidad del enlace*, *número de carriles*, *modos de transporte que pueden utilizar cada carril (bus, coche, taxi, etc)*, *densidad del tráfico*, etc.

Este modelo sencillo puede ser detallado añadiendo nodos para indicar los posibles giros permitidos en una intersección, tal y como se muestra en la figura 3.2.

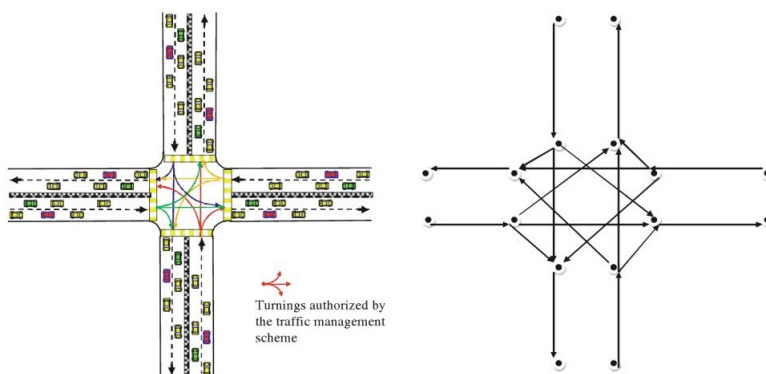


Figura 3.2: Representación de una intersección (Barceló 2010) [4]

Una vez que tenemos un modelo sencillo, al cual se le pueden añadir tantos detalles como deseemos en forma de atributos en los nodos y en los arcos, se nos presenta el problema de la búsqueda de caminos en una red viaria.

Como solución a dicho problema aparece el estudio *A new data structure to represent road networks* [5], en el cual se propone utilizar tres niveles de detalle para representar el mapa. Estos niveles pueden observarse en la figura 3.3 y en la tabla 3.1

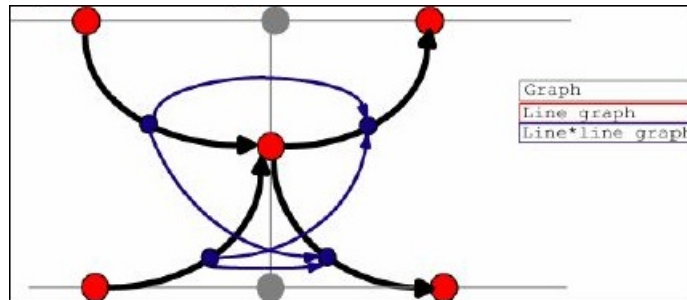


Figura 3.3: Representación de una intersección (Bogaert) [5]

Nivel	Nodos	Arcos
0 (un grafo)	Intersecciones	Segmentos de vía
1 (grafo de líneas)	Segmentos de vía	Giros
2 (grafo de líneas de líneas)	Giros	Conexiones entre giros

Tabla 3.1: Niveles representados en la figura 3.3

3.3 Sistemas de partículas conductistas

Los sistemas de partículas son colecciones de objetos independientes, a menudo representados por una sola forma o punto. Todas las partículas tienen una serie de atributos los cuales permiten tener distintos tipos de partícula, tanto en aspecto como en comportamiento.

Generalmente, la implementación de sistemas de partículas conductistas se lleva a cabo mediante la utilización de máquinas de estados finitos, las cuales sirven para modelar el comportamiento de las partículas, de manera que éstas actuarán de una forma u otra en base a sus atributos.

En el caso particular de este proyecto, cada vehículo será una partícula con una serie de características o atributos, lo que nos permitirá modelar distintos tipos de vehículo y distintos tipos de comportamiento de los conductores, con el fin de representar la variedad de los mismos que podemos observar en la realidad.

3.4 Modelos de redes viarias reales a partir de imágenes

La obtención de redes viarias a partir de imágenes de satélite es un campo de investigación muy importante debido a la gran cantidad de aplicaciones en las que se puede utilizar (militares, civiles, científicas, etc).

El problema consiste en detectar en una imagen, qué es una carretera (camino o vía) y qué no lo es, de forma que esta información pueda ser extraída y procesada.

En el artículo *Road network extraction and intersection detection from aerial images by tracking road footprints* [6] se describe un método de extracción de redes viarias que proporciona muy buenos resultados.

Este método consta de tres pasos:

1. *Automatic road seeding*: Se analiza el vecindario de todos los píxeles de la imagen y se decide si el pixel es un punto de inicio válido para un segmento de carretera.
2. *Road tracking*: Comenzando a partir de todas las semillas, se extienden los segmentos de carretera, iterativamente, en una, dos o más direcciones. Se analizan iterativamente el vecindario local de todos los puntos activos en la red viaria para decidir si parar, continuar en una dirección, o dividir en dos o tres direcciones. Este algoritmo va creando un árbol de carreteras.
3. *Road tree pruning*: Los pasos previos producen una red viaria que contiene casi todos los segmentos de carretera, pero sufre de sobreextracción y pérdidas. Los autores del artículo utilizan una regla de decisión bayesiana para eliminar trozos de la red extraída que parezcan no ser carreteras.

La inclusión de este método en el proyecto posibilitaría incluir redes viarias reales a la simulación, lo que permitiría probar distintas soluciones de ordenación viaria en entornos reales.

3.5 Herramientas gráficas

Capítulo 4

Propuesta

En este capítulo expondremos el trabajo que se pretende realizar para conseguir cada uno de los objetivos mostrados en el capítulo anterior.

4.1 Objetivo principal del proyecto

Se desarrollará una aplicación gráfica multiplataforma, mediante la herramienta *Unity* utilizando el motor gráfico que incluye, para alojar el simulador de tráfico; y se desarrollará una aplicación [TODO: ¿cómo?] para la obtención de mapas de carreteras reales.

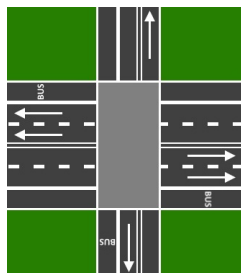
4.2 Red viaria

La red viaria que utilizará el simulador vendrá especificada en un fichero con formato GraphML [7]. Dicha red estará determinada por dos grafos, un grafo no dirigido, de topología, en el que los cruces serán los nodos, y las vías serán los arcos; y un grafo dirigido, para indicar los giros, en el que las vías serán los nodos y los giros serán los arcos. Debido a las restricciones que impone la definición de GraphML sobre los identificadores, cada grafo deberá estar en un fichero distinto. Para ello usaremos dos ficheros por cada red viaria, siendo los ficheros *.topology.graphml para el primer tipo de grafo y los ficheros *.turns.graphml para el segundo. El identificador del grafo en cada uno de los ficheros de la pareja de cada grafo será el mismo.

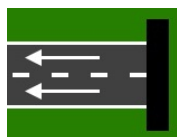
Nodos del grafo de topología

Debemos distinguir tres tipos:

- Nodos de cruce: Representan un cruce de vías típico. El grado de estos nodos es igual o superior a tres.



- Nodos de límite de vía: Representan el límite del área simulada en esa vía y le permitirán al usuario configurar el flujo de tráfico de esa vía (en el caso de que la vía asociada tenga algún carril entrante al área de simulación). El grado de este tipo de nodo es igual a uno.



- Nodos de continuación: Representan ángulos en las vías de tal forma que en ellos no se producirán intersecciones de vías. El grado de este tipo de nodo es igual a dos.



Por tanto, cada nodo estará definido por un identificador alfanumérico, el tipo de nodo (0: nodo de cruce, 1: nodo de límite de vía, 2: nodo de continuación), sus coordenadas en el plano bidimensional x,y, y para los nodos de tipo cruce, el tipo de cruce, dado que queremos dar la oportunidad al usuario de elegir entre cruce normal (0) o rotonda (1).

Arcos del grafo de topología

Cada arco estará definido por un identificador alfanumérico, el nodo de origen, el nodo de destino, una cadena de texto para poder dar nombre a la vía, y dos cadenas de texto que servirán para indicar los tipos de carril en cada sentido.

Cada una de las cadenas de tipo de carril especificará el tipo de carril o carriles de ese sentido que haya desde el exterior de la vía hacia el interior de la misma utilizando los códigos: P: para indicar un carril de transporte

público, N: un carril normal, A: un carril de aparcamiento, V: un carril de Bus/VAO, o la cadena '0' para indicar que no hay carriles en ese sentido.

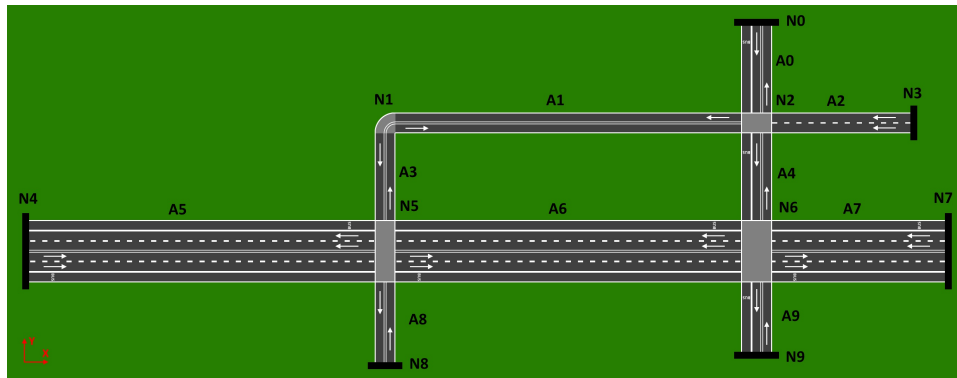
Nodos del grafo de giros

Cada nodo estará definido por un identificador alfanumérico que se corresponderá con uno de los identificadores de los arcos del grafo de topología.

Arcos del grafo de giros

Cada arco estará definido por un identificador alfanumérico, el nodo de origen, el nodo de destino. Si el arco está definido significa que el giro está permitido.

Ejemplo Para ilustrar esta especificación vamos a ver un ejemplo de red viaria y cuál sería su representación en GraphML.



Como se puede apreciar en la imagen, el grafo de topología asociado consta de diez nodos, tres nodos de tipo cruce, un nodo de tipo continuación y seis nodos de tipo límite de vía. Además, el grafo cuenta con diez arcos que representan las vías de la imagen.

Para este ejemplo se ha supuesto que en cada cruce se puede ir en todas las direcciones que admite esta red, el grafo de giros está compuesto por diez nodos, los arcos del grafo de topología, que en este ejemplo están todos presentes en el grafo debido a que todos están unidos a algún nodo de tipo cruce; y treinta y tres arcos que representan los giros.

A continuación se mostrará la representación de la red viaria con sintaxis de GraphML.

Grafo de topología:

```
<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
  <key id="node_type" for="node" attr.name="Tipo-de-nodo" attr.type="int">
    <default>0</default>
  </key>
  <key id="pos_x" for="node" attr.name="Coordenada_X" attr.type="double">
  <key id="pos_y" for="node" attr.name="Coordenada_Y" attr.type="double">
  <key id="intersection_type" for="node" attr.name="Tipo-de-cruce" attr.type="int">
    <default>0</default>
  </key>
  <key id="name" for="edge" attr.name="Nombre-de-la-via" attr.type="string">
    <default></default>
  </key>
  <key id="src_des" for="edge" attr.name="Vias_sentido_origen_destino" attr.type="string">
    <default>0</default>
  </key>
  <key id="des_src" for="edge" attr.name="Vias_sentido_destino_origen" attr.type="string">
    <default>0</default>
  </key>
  <graph id="ejemplo_topologia" edgedefault="undirected">
    <node id="n0"/>
    <data key="node_type">1</data>
    <data key="pos_x">2168</data>
    <data key="pos_y">1044</data>
    </node>
    <node id="n1"/>
    <data key="node_type">2</data>
    <data key="pos_x">1095</data>
    <data key="pos_y">751</data>
    </node>
    <node id="n2"/>
    <data key="pos_x">2168</data>
    <data key="pos_y">751</data>
    </node>
    <node id="n3"/>
    <data key="node_type">1</data>
    <data key="pos_x">2623</data>
    <data key="pos_y">751</data>
    </node>
    <node id="n4"/>
    <data key="node_type">1</data>
    <data key="pos_x">53</data>
    <data key="pos_y">381</data>
    </node>
    <node id="n5"/>
    <data key="pos_x">1095</data>
    <data key="pos_y">381</data>
    </node>
    <node id="n6"/>
    <data key="pos_x">2168</data>
    <data key="pos_y">381</data>
    </node>
    <node id="n7"/>
    <data key="node_type">1</data>
    <data key="pos_x">2723</data>
    <data key="pos_y">381</data>
    </node>
    <node id="n8"/>
    <data key="node_type">1</data>
    <data key="pos_x">1095</data>
    <data key="pos_y">7</data>
    </node>
    <node id="n9"/>
    <data key="node_type">1</data>
    <data key="pos_x">2168</data>
    <data key="pos_y">83</data>
    </node>
    <edge id="a0" source="n0" target="n2"/>
    <data key="src_des">PN</data>
    <data key="des_src">N</data>
    </edge>
    <edge id="a1" source="n1" target="n2"/>
    <data key="src_des">N</data>
    <data key="des_src">N</data>
    </edge>
    <edge id="a2" source="n2" target="n3"/>
    <data key="des_src">NN</data>
    </edge>
    <edge id="a3" source="n1" target="n5"/>
```



```

        <data key="src_des">N</data>
        <data key="des_src">N</data>
    </edge>
    <edge id="a4" source="n2" target="n6"/>
    <data key="src_des">PN</data>
    <data key="des_src">N</data>
    </edge>
    <edge id="a5" source="n4" target="n5"/>
    <data key="src_des">PNN</data>
    <data key="des_src">PNN</data>
    </edge>
    <edge id="a6" source="n5" target="n6"/>
    <data key="src_des">PNN</data>
    <data key="des_src">PNN</data>
    </edge>
    <edge id="a7" source="n6" target="n7"/>
    <data key="src_des">PNN</data>
    <data key="des_src">PNN</data>
    </edge>
    <edge id="a8" source="n5" target="n8"/>
    <data key="src_des">N</data>
    <data key="des_src">N</data>
    </edge>
    <edge id="a9" source="n6" target="n9"/>
    <data key="src_des">PN</data>
    <data key="des_src">N</data>
    </edge>
</graph>
</graphml>

```

Grafo de giros:

```

<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
    <graph id="ejemplo_giros" edgedefault="directed">
        <node id="a0"/>
        <node id="a1"/>
        <node id="a2"/>
        <node id="a3"/>
        <node id="a4"/>
        <node id="a5"/>
        <node id="a6"/>
        <node id="a7"/>
        <node id="a8"/>
        <node id="a9"/>
        <edge id="e00" source="a0" target="a1"/>
        <edge id="e01" source="a0" target="a4"/>
        <edge id="e02" source="a1" target="a0"/>
        <edge id="e03" source="a1" target="a4"/>
        <edge id="e04" source="a2" target="a0"/>
        <edge id="e05" source="a2" target="a1"/>
        <edge id="e06" source="a2" target="a4"/>
        <edge id="e07" source="a3" target="a5"/>
        <edge id="e08" source="a3" target="a6"/>
        <edge id="e09" source="a3" target="a8"/>
        <edge id="e10" source="a4" target="a0"/>
        <edge id="e11" source="a4" target="a1"/>
        <edge id="e12" source="a4" target="a6"/>
        <edge id="e13" source="a4" target="a7"/>
        <edge id="e14" source="a4" target="a9"/>
        <edge id="e15" source="a5" target="a3"/>
        <edge id="e16" source="a5" target="a6"/>
        <edge id="e17" source="a5" target="a8"/>
        <edge id="e18" source="a6" target="a3"/>
        <edge id="e19" source="a6" target="a4"/>
        <edge id="e20" source="a6" target="a5"/>
        <edge id="e21" source="a6" target="a7"/>
        <edge id="e22" source="a6" target="a8"/>
        <edge id="e23" source="a6" target="a9"/>
        <edge id="e24" source="a7" target="a4"/>
        <edge id="e25" source="a7" target="a6"/>
        <edge id="e26" source="a7" target="a9"/>
        <edge id="e27" source="a8" target="a3"/>
        <edge id="e28" source="a8" target="a5"/>
        <edge id="e29" source="a8" target="a6"/>
        <edge id="e30" source="a9" target="a4"/>
        <edge id="e31" source="a9" target="a6"/>
        <edge id="e32" source="a9" target="a7"/>
    </graph>
</graphml>

```

</graphml>

4.3 Señales de tráfico y peatones

4.4 Vehículos

Se diseñarán tres modelos con ruedas animadas mediante la herramienta
..... ¡¡¡¡ POR DEFINIR !!!!

4.5 Conductores

La lógica de los conductores se realizará mediante una máquina de estados.

4.6 Entorno

4.7 Uso de redes viarias reales

Bibliografía

- [1] D. Drew, *Traffic flow theory and control*. New York: McGraw-Hill, 1968.
- [2] M. Pursula, “Simulation of traffic systems - an overview,” *Journal of Geographic Information and Decision Analysis*, vol. 3, no. 1, pp. 1–8, 1999.
- [3] H. Kallberg, “Traffic simulation,” Master’s thesis, University of Technology, Transportation Engineering, Espoo, 1971.
- [4] J. Barceló, ed., *Fundamentals of Traffic Simulation*, vol. 145 of *International Series in Operations Research & Management Science*. Springer New York, 2010.
- [5] P. Bogaert, V. Geografie, N. V. D. Weghe, V. Geografie, R. Maddens, V. Geografie, L. D. Temmerman, V. Geografie, and V. Geografie, “A new data structure to represent road networks.”
- [6] J. C. F. M. C. P. W. Jiuxiang Hu, Anshuman Razdan, “Road network extraction and intersection detection from aerial images by tracking road footprints,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, p. 14, Diciembre 2007.
- [7] GraphML Team, *The GraphML File Format*. graphml.graphdrawing.org, 2001-2007.