

**UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA
UNAN-LEÓN**



**FACULTAD DE CIENCIAS Y TECNOLOGÍA
DPTO DE MATEMÁTICA Y ESTADÍSTICA**

**TESIS PARA OPTAR AL TÍTULO DE
LICENCIADO EN MATEMÁTICA**

TEMA:

“SIMULACIÓN DE TRÁFICO EN EL CENTRO DE LA CIUDAD DE LEÓN”

Presentado por:

Br. Amilcar Geovanny Cabrera Ramírez
Br. Elí Saúl Ponce
Br. Henrys David Rivas Salinas

Tutor: Dr. Ramiro José Cáceres Espinoza

León, agosto del 2013

“A la libertad por la Universidad”



Simulación de tráfico en el centro de la ciudad de León

AGRADECIMIENTO

A Dios todo poderosos por brindarnos la oportunidad de vivir y culminar nuestros estudios con la sabiduría y la fortaleza necesaria para superar cualquier obstáculo y emprender nuevos caminos.

A nuestro tutor Dr. Ramiro José Cáceres Espinoza quien además de ser uno de nuestros principales formadores se dio a la tarea de guiar nuestro trabajo de investigación con paciencia, sabiduría y mucho empeño.

A los docentes del Departamento de Matemática y estadística que nos transmitieron sus conocimientos durante toda nuestra formación profesional.

A todas las personas que de una u otra manera influyeron considerablemente en nuestro desarrollo profesional.

DEDICATORIA

A mi madre **Martha de los Ángeles Cabrera** quien con su amor, apoyo, dedicación, esfuerzo y confianza me motivo a continuar el camino de mi formación profesional.

A mis familiares que en ningún momento me dejaron solo por muy difícil que se viera el camino.

A mis amigos en los que siempre encontré palabras de motivación y aliento.

Amilcar Geovanny Cabrera Ramírez.

DEDICATORIA

“Los hombres geniales empiezan grandes obras, los hombres trabajadores las terminan”

Leonardo da Vinci

A mi madre **María Auxiliadora Ponce Espinoza** y mis hermanos **Carlos Emilio, Edwing Javier, José Luis y María José.**

Elí Saúl Ponce.

DEDICATORIA

A Dios nuestro padre celestial por ser el a quien le debemos nuestra vida, la fuerza, la sabiduría y el entendimiento.

A mis padres:

Luisa Amelia Salinas Aguilar

Luis Adolfo Rivas Gutiérrez

Y Hermanos

Por ser ejemplos de amor, abnegación, empeño y sacrificio para que lograra culminar mis estudios

Al grupo **Casa Nicaragua** por haber depositado su confianza y brindarme su apoyo económico, moral e intelectual.

A profesora **Isabel Vásquez** por ser un eje fundamental y una guía para mi desarrollo.

A los maestros que con mucho empeño hicieron posible el éxito en mi vida profesional.

A mis compañeros y amigos que también colaboraron en mi formación.

Henry David Rivas Salinas.

ÍNDICE

1. Introducción	1
2. Antecedentes	3
3. Objetivos	5
4. Aspectos teóricos.....	6
4.1. Simulación	6
4.1.1. Etapas de la simulación	6
4.1.2. Clasificación de los modelos de simulación	8
4.2. Tráfico vehicular	9
4.2.1. Parámetros esenciales en el estudio del tráfico vehicular	9
4.2.2. Reseña de los modelos de tráfico.....	11
4.2.3. Algunos modelos de enfoque macroscópico	13
4.2.4. Modelos de enfoque microscópicos.....	14
4.2.5. Congestionamiento vehicular.....	15
4.3. Programación orientada a objetos en c++	16
4.3.1. Historia del lenguaje c/c++.....	16
4.3.2. Características.....	17
4.3.3. Orientación a objetos	23
4.4. Teoría de colas	26
4.4.1. Medidas de rendimiento de un sistema de colas	27
4.4.2. Clasificación de los modelos de cola según Kendall.....	27
4.4.3. Modelos de colas más comunes.....	28
5. La simulación de tráfico	31
5.1. Construcción de la red.....	31
5.2. El trabajo de campo.....	33
5.3. Medidas de desempeño de la red de tráfico.....	33
5.4. Caracterización de los elementos de la simulación	34
5.4.1. Circulación de los vehículos en la red de tráfico	36
5.4.2. Ingreso y salida de la red de tráfico	36
5.4.3. Los semáforos	37
5.4.4. El cruce de las intersecciones.....	39
5.4.5. Clases y funciones en la simulación	39
5.4.6. Entrada de la simulación.....	47

5.4.7.	Salida de la simulación (resultados)	49
5.4.8.	Visualización de la simulación.	49
5.5.	La validación de la simulación	51
5.5.1.	Medidas de desempeño en base a la simulación.	51
5.5.2.	Medidas de desempeño en base a datos reales	53
5.5.3.	Cálculo de medidas de desempeño reales para la vía 16.	54
5.5.4.	Comparación de medidas de desempeño	56
6.	Conclusiones	64
7.	Recomendaciones	65
8.	Bibliografía	66
9.	Anexos	67
9.1.	Datos de frecuencias de giro en la red	67
9.2.	Frecuencias de giro en los nodos de la red	72
9.3.	Tiempos de entrada, llegada y salida de los autos en la vía 16.....	73

1. INTRODUCCIÓN

Si salimos a la calle y tomamos un tiempo para observar nos daremos cuenta que las personas constantemente se desplazan de un lugar a otro para desarrollar sus actividades diarias (Ir al trabajo, la escuela, comprar o vender mercancía etc.) para ello hacemos uso de vehículos, calles, semáforos, avenidas, puentes peatonales entre otros, en fin un sistema formado por todos esos parámetros que continuamente están interactuando.

El tráfico vehicular es un fenómeno complejo de nuestra vida diaria; la interacción de todos los actores involucrados (calles, semáforos, peatones, etc.) es bastante difícil de analizar y a menudo son estudiados de manera individual, sin embargo mediante simulación se han desarrollado modelos para lograr una mayor comprensión sobre el funcionamiento del sistema completo.

La simulación de un sistema consiste en crear un modelo de un sistema real y sobre el aplicar técnicas de análisis para hacer conjeturas sobre el funcionamiento de éste; El modelo creado permite hacerle cambios para mejorar su eficiencia, cambios que no pueden realizarse mediante el método de prueba y error en el mismo sistema, pues podrían conducir al caos en él. El simulador le permite observar como ocurren embotellamientos sin causa aparente, el efecto de la incorporación de una vía al sistema, el bloqueo de una calle, etc. El interesado podría alterar los parámetros y observar el efecto del cambio.

Un sistema de tráfico eficiente es esencial para el desarrollo económico de la sociedad y la calidad de vida humana. El aumento en el parque vehicular en el centro de la ciudad de León en los últimos años, la falta de regulación de las zonas de parqueo, el irrespeto a las reglas de tránsito etc. ha provocado el problema del congestionamiento vehicular el cual nos afecta a todos. Estos problemas se manifiestan en el riesgo de sufrir accidentes de cualquier índole.

Podemos caracterizar el sistema de tráfico mediante parámetros tales como la cantidad de semáforos peatonales y vehiculares, la fluidez del tráfico, la capacidad de estacionamiento, forma y frecuencia de ingreso de los vehículos, la velocidad de circulación, etc. El problema puede verse entonces como el del desconocimiento de las interrelaciones de las partes del sistema, como influye cada una de ellas sobre las demás y sobre el sistema completo. Hasta el momento no se ha explorado la forma de investigar las interrelaciones entre las partes de este sistema (como influye el declarar una calle como peatonal, poner un semáforo en una esquina, declarar áreas de parqueo en una calle, cambiar el sentido de la circulación en una calle, etc.). Nos interesa descubrir los cambios al sistema que conduzcan a mejorar la situación planteada en términos de los parámetros del sistema.

Para dar inicio al estudio del sistema de tráfico del centro de León realizamos exploraciones (observaciones), para darnos una idea acerca de la estructura de éste y la interacción de sus actores. Posteriormente dedicamos tiempo para el estudio de la teoría de tráfico, así como de programación en C++ y programación orientada a objetos como herramienta principal en la construcción de los programas que implementan nuestro modelo; luego realizamos trabajo de campo para la georeferenciación, señalización de tránsito y recolección de frecuencias de giro del flujo vehicular en las intersecciones (nodos) comprendidas en el área limitada. Para la validación del modelo de tráfico nos auxiliamos de los sistemas de colas, tomando algunas de sus medidas de rendimiento para ser usadas en el análisis de la eficiencia del sistema real. Finalmente analizamos resultados y damos algunas observaciones sobre todo el trabajo realizado.

2. ANTECEDENTES

Desde mucho tiempo atrás se han realizado estudios sobre el flujo vehicular aplicando diferentes teorías y procedimientos, como es el caso de la probabilidad, flujo de ondas y teoría de colas; sin embargo no se ha llegado a una idea general que satisfaga todas las necesidades de cualquier sistema en concreto.

Greenshield inicia el estudio del flujo vehicular modelando el comportamiento de este en una sola vía de tráfico, proponiendo una relación lineal entre velocidad y densidad.

Más tarde, con el incremento poblacional y el aumento del número de vehículos, en la década de los cincuenta se retoma la idea de cómo estudiar el comportamiento del flujo vehicular, surgiendo el llamado modelo de Car-Following, teoría de ondas de tráfico, y se da una implementación más formal a la teoría de colas para la descripción del flujo vehicular.

Posteriormente los investigadores Ruschel y Pipes proponen un modelo de tráfico del tipo microscópico, que describe el comportamiento detallado entre los vehículos en un solo carril de tal forma que el espacio entre ellos sea el más mínimo posible.

Seguidamente Lighthill y Whitman proponen un modelo macroscópico, modelando el tráfico como un flujo continuo. Publican una de las primeras teorías de modelado macroscópico de flujo de tráfico en carreteras.

Hasta éste momento los investigadores habían prestado atención al estudio de los modelos de aceleración en régimen de seguimiento de vehículos, fue entonces cuando surge los modelos de aceleración libre, estos son empleados para aquellos casos donde el conductor no está influido por un vehículo precedente, como por ejemplo cuando actúan en función de semáforos, obstáculos o bien se mueve con completa libertad.

Durante todo este tiempo los estudios anteriores han servido para mejorar las ideas sobre la descripción del tráfico vehicular; el estudio más reciente realizado por los científicos Kai Nagel y Michael Schreckenberg, ellos proponen un modelo de flujo de tráfico con autómatas celulares, donde cada célula del autómata equivale a un vehículo en movimiento.

En el caso de Nicaragua no conocemos acerca de algún estudio sobre modelos de tráfico que sirva para la regulación del tráfico vehicular, podemos decir que las instituciones encargadas realizan este proceso de forma intuitiva o por conveniencia, es decir, se distribuyen las vías y las señales de tránsito a la manera que ellos creen correcto o donde se consideran necesarias.

El objetivo de una simulación de tráfico validada para el centro de la ciudad de León, es dar una forma no empírica de administración del tráfico vehicular, de tal manera que se puedan implementar mejores herramientas para el buen funcionamiento del sistema, lo cual resultara beneficioso para todos.

3. OBJETIVOS

OBJETIVO GENERAL:

- ❖ Aplicar conocimientos matemáticos que brinden soluciones a los problemas presentados por la actual red vial en el centro de la ciudad de León.

OBJETIVOS ESPECIFICOS:

- ❖ Elaborar una breve descripción de las características más importantes sobre algunos de los actores involucrados en el sistema de tráfico vehicular del centro de la ciudad de León.
- ❖ Mostrar el funcionamiento y la interacción del sistema de tráfico vehicular de la ciudad de León.

4. ASPECTOS TEÓRICOS

A continuación presentamos una serie de aspectos teóricos con el fin de familiarizarnos con la terminología empleada en el desarrollo de nuestro trabajo. Son conceptos relacionados con simulación, tráfico vehicular, programación c++ y sistemas de cola.

4.1. SIMULACIÓN

Un modelo es un bosquejo que representa un sistema real con cierto grado de precisión y en la forma más completa posible, esto es, debe representar la realidad de manera simplificada. Los modelos son muy útiles para comparar, predecir, describir, explicar o comprender mejor un fenómeno de la realidad, cuando es difícil o costoso trabajar directamente en la entidad misma.

4.1.1. ETAPAS DE LA SIMULACIÓN

Antes de continuar presentaremos tres ideas que describen con mucha claridad el concepto de simulación:

- Simulación es la experimentación con un modelo de una hipótesis o un conjunto de hipótesis de trabajo.
- Simulación es una técnica numérica para conducir experimentos en una computadora digital. Estos experimentos comprenden ciertos tipos de relaciones matemáticas y lógicas, las cuales son necesarias para describir el comportamiento y la estructura de sistemas complejos del mundo real a través de largos periodos de tiempo".
- Según R.E. Shannon, simulación es el proceso de diseñar un modelo de un sistema real y llevar a término experiencias con él, con la finalidad de comprender el comportamiento del sistema o evaluar nuevas estrategias dentro de los límites impuestos por un cierto criterio o un conjunto de ellos, para el funcionamiento del sistema.

Etapas para realizar un estudio de simulación

Definición del sistema

Consiste en estudiar el contexto del problema, identificar los objetivos del proyecto, especificar los índices de medición de la efectividad del sistema, especificar los objetivos específicos de la modelación y definir el sistema que se va a modelar. Esta actividad requiere de la conversación con las personas que participan de las actividades que se están analizando mediante el modelo.

Formulación del modelo

Una vez definidos con exactitud los resultados que se esperan obtener del estudio, se define y construye el modelo con el cual se obtendrán los resultados deseados.

En la formulación del modelo es necesario definir todas las variables que forman parte de él, sus relaciones lógicas y los diagramas de flujo que describan en forma completa el modelo.

Colección de datos

Es importante que se definan con claridad y exactitud los datos que el modelo va a requerir para producir los resultados deseados.

Implementación del modelo en la computadora

Con el modelo definido, el siguiente paso es decidir si se utiliza algún lenguaje de programación, o se utiliza algún paquete para procesarlo en la computadora y obtener los resultados deseados.

Validación

A través de esta etapa es posible detallar deficiencias en la formulación del modelo o en los datos alimentados al modelo. Las formas más comunes de validar un modelo son:

- La opinión de expertos sobre los resultados de la simulación.

- La exactitud con que se predicen datos históricos.

- La exactitud en la predicción del futuro.

- La comprobación de falla del modelo de simulación al utilizar datos que hacen fallar al sistema real.

- La aceptación y confianza en el modelo de la persona que hará uso de los resultados que arroje el experimento de simulación.

Experimentación

La experimentación con el modelo se realiza después que éste haya sido validado. La experimentación consiste en generar los datos deseados y en realizar un análisis de sensibilidad de los índices requeridos.

Interpretación

En esta etapa del estudio, se interpretan los resultados que arroja la simulación y con base a esto se toma una decisión. Es obvio que los resultados que se obtienen de un estudio de simulación ayudan a soportar las decisiones que pueda sugerir el estudio.

Documentación

Dos tipos de documentación son requeridos para hacer un mejor uso del modelo de simulación. La primera se refiere a la documentación del tipo técnico y la segunda se refiere al manual del usuario, con el cual se facilita la interacción y el uso del modelo desarrollado.

Simulación por computadora

Es un intento de modelar situaciones de la vida real por medio de un programa de computadora, lo que requiere ser estudiado para ver cómo es que trabaja el

sistema. Ya sea por cambio de variables, quizás predicciones hechas acerca del comportamiento del sistema.

La simulación por computadora se ha convertido en una parte útil del modelado de muchos sistemas naturales en física, química y biología, y sistemas humanos como la economía y las ciencias sociales

Tradicionalmente, el modelado formal de sistemas ha sido a través de un modelo matemático, que intenta encontrar soluciones analíticas a problemas que permiten la predicción del comportamiento de un sistema de un conjunto de parámetros y condiciones iniciales. La simulación por computadora es frecuentemente usada para sistemas cuya solución analítica sería prohibitiva o imposible.

4.1.2. CLASIFICACIÓN DE LOS MODELOS DE SIMULACIÓN

Hoy en día con el desarrollo de nuevas tecnologías (paquetes de computadora y software) y la necesidad de estudiar diversos fenómenos existen distintos modelos para simular cualquier fenómeno, cada uno con características particulares y enfoques diferentes. Los podemos clasificar como:

- Modelos continuos

Son sistemas cuyas variables cambian de manera continua con el transcurso del tiempo, por lo general se representan aplicando ecuaciones diferenciales que describen las interacciones entre los diferentes elementos del sistema. Un ejemplo característico es el estudio de la dinámica demográfica mundial.

- Modelos Discretos

Consiste en la modelación de un sistema por medio de una representación en la cual el estado de las variables cambia instantáneamente en intervalos de tiempos separados. (En términos matemáticos el sistema solo puede cambiar en instante de tiempo contables).

Un ejemplo típico de simulación discreta ocurre en los sistemas de colas donde estamos interesados en la estimación de medidas como el tiempo de espera promedio o la longitud de la línea de espera.

Otro ejemplo es la simulación del tráfico en una carretera caracterizando el movimiento de los vehículos de forma individual.

Experimentación del proceso de Simulación.

En el análisis de sistemas se procura comprender la forma de operación de un sistema ya existente o planteado. Construyendo un modelo, aun cuando lo ideal sería experimentar sobre el propio sistema, y mediante la simulación, se indaga en el comportamiento de este.

En el diseño de sistemas lo que se busca es generar un sistema que satisfaga ciertas restricciones. En este caso, el diseñador puede seleccionar o plantear determinados componentes de sistemas, y conceptualmente elige una determinada combinación de componentes para construir el modelo del sistema. El sistema propuesto se modela y se predice su comportamiento a partir del conocimiento del mismo; luego se compara el comportamiento pronosticado con el comportamiento deseado, en caso que la comparación sea aceptable, se aprueba el diseño, de lo contrario, se rediseña el sistema y se vuelve a repetir el proceso.

4.2. TRÁFICO VEHICULAR

El sistema de tráfico vehicular se describe como un sistema complejo y difícil de modelar, debido a que interviene el factor humano: los conductores responden de diversas maneras a las condiciones del tráfico en torno a ellos, un ejemplo claro es la reacción que tiene un conductor cuando el auto delantero frena, él puede optar por rebasar o detenerse y esperar que avance.

Definición (tráfico vehicular)

Formalmente podemos decir que es el fenómeno causado por el flujo de vehículos en una vía, calle o autopista. De manera similar se presenta en otros fenómenos como el flujo de partículas y el de peatones. También podemos relacionar este proceso con la hidrodinámica de fluidos, tratando al tránsito como un fluido unidimensional de izquierda a derecha.

4.2.1. PARÁMETROS ESENCIALES EN EL ESTUDIO DEL TRÁFICO VEHICULAR

Para describir la circulación de los vehículos en una vía, carretera o autopista es necesario definir los siguientes parámetros:

Velocidad

Sabemos que los vehículos en una vía por lo general están en constante movimiento, debido a este fenómeno la velocidad de cada vehículo puede estar variando, a continuación presentamos algunas definiciones sobre la velocidad de un vehículo.

Definición (velocidad)

La velocidad es la distancia recorrida por un vehículo durante una unidad de tiempo regularmente se expresa en kilómetros por hora (Km/h).

Definición (velocidad media de recorrido)

La velocidad media es el promedio de las velocidades de todos los vehículos que pasan por un punto fijo de la vía durante cierto periodo de tiempo.

Definición (velocidad media espacial, V_e)

La velocidad media espacial es la media aritmética de las velocidades de todos los vehículos que están en un tramo de vía en un momento dado.

Definición (velocidad media espacial a flujo libre)

Es la velocidad media de los vehículos cuando presentan volúmenes bajos de tráfico, y no hay imposición de restricciones de sus velocidades, ni por interrupción vehicular ni por regulaciones del tráfico.

Definición (capacidad de la vía)

La capacidad es el máximo número de vehículos que pueden circular en un punto dado durante un período específico de tiempo, bajo condiciones predeterminadas de la carretera y el tránsito.

Intensidad de Tráfico

Definición (intensidad de tráfico)

Se dice que es el número de vehículos que pasa a través de una parte o sección fija de una vía o carretera por unidad de tiempo. Esta dada por la ecuación:

$$I = N/T$$

Donde N es el número de vehículos que pasan por la sección de la vía o carretera y T es el intervalo de tiempo determinado.

Podemos calcular la intensidad mediante conteos manuales en determinados secciones de la vía; sin embargo, se han adoptado herramientas electrónicas (aparatos contadores, cámaras de videos) posiblemente más eficaces.

Observación:

La intensidad de tráfico es el factor más influyente sobre la velocidad media en una vía. Se puede señalar que mientras la intensidad de tráfico es baja, los conductores pueden mantener la velocidad que ellos consideran más adecuada, mientras que cuando aumenta la intensidad la velocidad de cada conductor viene determinada en gran parte por la de los demás, produciéndose una disminución de la velocidad media. Cuando esta intensidad es muy alta y la carretera llega a estar congestionada, la velocidad resulta poco influida por las características de la vía e incluso es casi la misma para todo tipo de vehículos.

Densidad de Tráfico

Definición (densidad de tráfico)

La densidad de tráfico (k) es el número de vehículos que ocupan una longitud o área determinada de una vía o carretera en un momento dado. Esta dada por la

ecuación: $K = \frac{N}{d}$

Dónde:

N : Número de vehículos.

d : Longitud del tramo de vía.

Note que la densidad es una variable que determina la valoración que hacen los conductores de la calidad de la circulación en una vía.

Intervalos de tiempo entre vehículos

Definición (intervalo simple)

Llamamos intervalo simple al intervalo de tiempo que transcurre entre el paso por un punto fijo de la vía de los vehículos consecutivos i -ésimo y $(i+1)$ -ésimo.

Definición (intervalo promedio)

Llamamos intervalo promedio de un grupo de vehículos que pasan por un punto fijo de una vía al promedio h de todos los intervalos simples resultantes entre los pares de vehículos consecutivos que circulan por la vía.

Espaciamiento entre vehículos

Definición (espaciamiento simple)

Es la distancia que existe entre dos vehículos consecutivos en una vía, medida desde sus defensas traseras o delanteras. El espaciamiento simple es la suma de la distancia entre los vehículos y la longitud del vehículo.

Definición (espaciamiento promedio)

El espaciamiento promedio de un conjunto de vehículos que circulan por la vía en un momento dado, es el promedio de todos los espaciamientos simples existente entre los pares de vehículos consecutivos en la vía.

4.2.2. RESEÑA DE LOS MODELOS DE TRÁFICO

Sabemos que el tráfico vehicular es un fenómeno de naturaleza estocástico y multivariante, por ello es necesario estudiar los diversos modelos de tráfico de tal forma que podamos identificar sus características y naturaleza de estudio.

Para ello haremos uso de la siguiente clasificación

Según la medida del flujo vehicular

- Modelos macroscópicos

Su naturaleza de estudio es el flujo de vehículos a gran escala, es decir, describen el comportamiento del sistema como un solo agente sin tomar en cuenta en detalle los actores que lo componen de manera individual.

El fenómeno es visto como un solo conjunto, son modelos continuos caracterizando el proceso a través de ecuaciones diferenciales, su enfoque son las relaciones globales del flujo de tráfico.

En particular los parámetros descritos por estos modelos son: La densidad de tráfico, velocidad de los vehículos e intensidad del tráfico.

- Modelos microscópicos

Son modelos del tipo discreto, se enfocan en la descripción del comportamiento del flujo del tráfico vehicular a través de las entidades individuales y atómicas que interactúan unas con otras, en este caso cada vehículo individual. También considera los espaciamientos entre los vehículos individuales, así como sus velocidades.

Entre los modelos microscópicos tenemos los modelos de carro siguiente y los modelos con autómatas celulares.

- Modelos mesoscópicos

También llamados modelos cinéticos, su especialidad es el estudio de los autos por grupos, los clasifican de acuerdo a alguna categoría como por ejemplo según sus velocidades. En estos modelos se define una función que expresa la probabilidad de que un vehículo a una velocidad especificada durante cierto tiempo se encuentre en una determinada posición. Utilizan por lo general métodos de la mecánica estadística.

En función del tiempo

- Modelos continuos: si las variables cambian de modo continuo en función del tiempo.
- Modelos discretos: si las variables varían en un conjunto contable de instantes de tiempo.

En función del tipo de variables

- Modelos determinísticos: si en la representación no contiene variables aleatorias y todas sus variables se definen mediante relaciones exactas.
- Modelos estocásticos: si contiene al menos una variable no determinista.

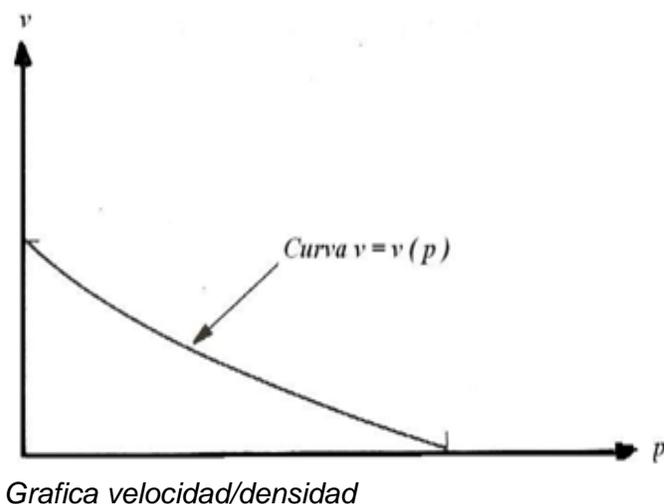
4.2.3. ALGUNOS MODELOS DE ENFOQUE MACROSCÓPICO

Modelo de Greenshield

Greenshield desarrolló uno de los primeros modelos para describir el tráfico, en este se estudia la relación entre la velocidad y densidad, también entre la velocidad y el flujo. Postuló que existe una relación lineal decreciente entre la velocidad y la densidad.

Observemos la correspondencia en la siguiente gráfica:

- 1) A mayor densidad menor velocidad de los vehículos.
- 2) A menor densidad mayor velocidad de los vehículos.



En la gráfica podemos observar como el vehículo mantiene una velocidad máxima cuando la densidad es mínima, y mientras más aumenta la densidad se va disminuyendo la velocidad.

Este hecho lo podemos ver de manera intuitiva cuando pensamos que un grupo de vehículos que viajan a mayor velocidad, viajan más dispersos, los vehículos conservan mayor distancia entre ellos y cuando la velocidad de estos vehículos disminuye, los vehículos se van juntando más, porque el avance es más lento.

La ecuación fundamental del modelo es:

$$V_e = V_l - (V_l/K_c) * K$$

V_e : Es la velocidad media espacial (km/h)

K : Es la densidad (veh/km/carril)

V_l : Velocidad media espacial a flujo libre (km/h)

K_c : Densidad de congestión según la capacidad de la vía (veh/km)

Modelo logarítmico de H. Greenberg

Este modelo se ha apoyado en la teoría de describir el flujo vehicular como el flujo hidrodinámico de partículas, ha tenido mayor repercusión en la ingeniería y el estudio del tráfico. De la combinación y discretización de las ecuaciones de movimiento y continuidad de fluidos compresibles aplicadas al flujo vehicular se obtienen las principales ecuaciones. Debemos tener en cuenta que el modelo, tiene gran nivel de validez especialmente en flujos congestionados, pero que falla en condiciones de flujo libre.

Su ecuación es:

$$V_e = V_m * \ln(K_c) / K$$

Dónde:

V_e : Es la velocidad media espacial (km/h)

K : Es la densidad (veh/km/)

V_m : Velocidad media espacial a flujo libre (km/h)

K_c : Densidad de congestión según la capacidad de la vía (veh/km)

Podemos ver en la ecuación anterior se conserva la notación del modelo de Greenshield, además:

V_m : Es la velocidad a flujo máximo sobre la capacidad de la vía.

Modelo exponencial de R. T. Underwood

Este modelo busca representar más fielmente el comportamiento del flujo libre y tiene la característica de no reflejar la velocidad igual a cero para las altas densidades (condición de congestión) por esta razón los parámetros del modelo son: densidad a intensidad máxima y velocidad a flujo libre.

4.2.4. MODELOS DE ENFOQUE MICROSCÓPICOS

Autómata Celular (AC)

Los AC son sistemas dinámicos discretos en todas sus componentes: espacio, tiempo y variables de estado, son útiles para modelar fenómenos dinámicos complejos, que se modelan normalmente a través de ecuaciones en derivadas parciales, evolucionan en el tiempo en base a reglas locales, describen los

sistemas de tráfico como una malla de células de igual tamaño; el tamaño de las células es escogido de tal forma que un vehículo con velocidad igual a otro se puede mover a la próxima célula durante un paso de tiempo discreto.

Modelo de Nagel-Schreckenberger

Es un modelo basado en autómatas celulares. Un modelo AC describe de manera discreta los movimientos de vehículos de célula a célula, un autómata celular origina el movimiento de los carros. El vehículo puede asumir una velocidad que va de un rango de valores discretos de cero a velocidad máxima.

En este modelo se observan en detalle los siguientes procesos: aceleración, desaceleración, proceso estocástico de desaceleración.

Modelo “Car-Following” (sigue al primero)

Fue desarrollado buscando estimar el comportamiento de dos vehículos en un flujo de tráfico, es decir, estimar la respuesta del vehículo con respecto al comportamiento del vehículo que viaja adelante (vehículo predecesor).

Describen la aceleración de coches “seguidores” como funciones de la distancia entre el líder y el seguidor, la velocidad relativa de los dos coches, y el tiempo de la reacción del conductor del coche seguidor.

La ecuación básica de los modelos Car-Following para una línea de tráfico está dada por:

$$Respuesta = Sensibilidad * Estímulo$$

Esta ecuación es fundamental para la descripción del proceso de seguida, y representa la ocurrencia de un hecho como estímulo que percibe un conductor y que lo anima a actuar, ya que dependiendo del grado de sensibilidad o advertencia del conductor en ese instante, se produce una reacción un instante después.

La respuesta está siempre representada por la aceleración o desaceleración del vehículo siguiente y el estímulo puede ser la distancia entre el vehículo siguiente y el vehículo de en frente, la velocidad relativa entre los dos vehículos o la aceleración relativa. La respuesta dependerá de la sensibilidad del conductor a un estímulo dado.

4.2.5. CONGESTIONAMIENTO VEHICULAR

La mayoría de las veces los congestionamientos viales son provocados por los malos hábitos de manejo de los conductores, algunas veces por accidentes, reparaciones de las vías o carreteras, y el mal funcionamiento de semáforos etc.

La congestión vehicular o vial, ya sea, urbana como interurbana, se refiere a la condición de un flujo vehicular que se va saturando debido al exceso de demanda de las vías o carreteras, produciendo incremento en los tiempos de viaje y atascamiento. Este fenómeno normalmente se produce en las horas punta u horas pico, y resulta frustrante para los conductores, ya que se ocasionan pérdidas de tiempo, y por consiguiente, consumo excesivo de combustible debido a que se está inactivo por mucho tiempo en un mismo lugar sin avanzar en el trayecto de un punto a otro.

Hay una serie de circunstancias específicas que causan o agravan la congestión, como el aumento del número de vehículos necesarios para un determinado caudal de personas o mercancías.

Es difícil predecir bajo qué condiciones se puede producir un embotellamiento pues puede ocurrir de repente. Si en determinada hora ocurre un incidente, por ejemplo un conductor que pierde el control de su auto y choca con otro en plena vía, puede tener repercusiones (un accidente en cadena), que luego se difunde y crea un embotellamiento de tráfico.

4.3. PROGRAMACIÓN ORIENTADA A OBJETOS EN C++

Un lenguaje de programación es una serie de comandos que nos permiten codificar instrucciones de manera que sean entendidas y ejecutadas por una computadora.

Existen varios lenguajes que permiten escribir un programa orientado a objetos y entre ellos C++.

4.3.1. HISTORIA DEL LENGUAJE C/C++

C++ es un lenguaje de programación de propósito general basado en el lenguaje de programación C. Nació en los laboratorios Bell de AT&T (Dennis Ritchie 1972), ha sido muy vinculado con el sistema operativo UNIX por su desarrollo en este sistema.

Este lenguaje ha evolucionado paralelamente a UNIX, ya que en 1980 se añaden al lenguaje C características t clases, chequeo del tipo de los argumentos de una función y conversión, si es necesaria, de los mismos, así como otras características resultando el llamado C con clases.

En 1983-1984 C con clases fue rediseñado, extendido y nuevamente implementado. El resultado se denominó C con clases. Las extensiones fueron principalmente funciones virtuales, funciones sobrecargadas, y operadores sobre cargados. C++ quedo disponible en 1995, fue creado por Bjarne Stroustrup.

Posteriormente, C++ ha sido ampliamente revisado y refinado, lo que ha dado lugar a añadir nuevas características, como herencia múltiple, funciones miembro `static` y `const`, miembros `protected`, tipos genéricos de datos o plantillas y manipulación de excepciones todo esto para convertir a C++ en un lenguaje más propicio para la escritura y utilización de bibliotecas.

Seguidamente se realizó una estandarización de tal forma que C++ sea estable, y que el código se pueda portar entre diferentes plataformas. Producto de esto, la biblioteca que incorpora actualmente C++ está escrita con la intención de incluir sólo aquellas clases que realmente son utilizadas por la mayoría de los programadores. Las facilidades proporcionadas por esta biblioteca estándar las podemos resumir como sigue:

- Soporte básico, por ejemplo: identificación del tipo de los objetos durante la ejecución y gestión de memoria.
- Soporte proporcionado por la biblioteca de C, por ejemplo: manipulación de cadenas, ficheros, etc.
- La clase `string` para la manipulación de cadenas de caracteres.
- Clases para la entrada y salida.
- Clases contenedor como vectores, listas y mapas.
- Algoritmos de búsqueda y de ordenación.
- Clases para trabajar con números como son: `cmath`, `complex`, y `cstdlib`.

Por tanto podemos decir que C++ es un lenguaje híbrido que ha adoptado las características de la programación orientada a objetos que no disminuyan su efectividad, y por otra parte, mejora sustancialmente las capacidades de C. Esto, junto con la biblioteca de clases soportada dota a C++ de una potencia, eficacia y flexibilidad que lo convierten en un estándar dentro de los lenguajes de programación orientada a objetos.

4.3.2. CARACTERÍSTICAS

A continuación mencionamos algunas características básicas de C++

Tipos de Datos

Es necesario especificar los tipos de datos que intervienen en una operación, así como el tipo del resultado para que el compilador reconozca la operación realizada. Los tipos de datos en C se clasifican en: tipos primitivos y tipos derivados.

Tipos primitivos

Se llaman primitivos porque ya están definidos por el compilador y los podemos clasificar como: tipos enteros (char, short, int, long y enum) y tipos reales (float y double).

Cada tipo primitivo tiene un rango diferente de valores positivos y negativos. El tipo de dato que se seleccione para declarar las variables de un determinado programa dependerá del rango y del tipo de valores que deberán almacenar cada uno de ellas y de si estos son enteros o fraccionarios, he aquí una caracterización de algunos de ellos.

- Char: se utiliza para declarar caracteres y datos enteros comprendidos entre -128 y +127, se define como un conjunto de 8 bits.
- Short: se utiliza para declarar datos enteros comprendidos entre -32768 y +32767, se define como un dato de 16 bits de longitud.
- Int: se utiliza para declarar datos enteros comprendidos entre -2147483648 y +2147483647, se define como un dato de 32 bits de longitud.
- Long: se utiliza para declarar datos enteros comprendidos entre -2147483648 y +2147483647 se define como un dato de 32 bits de longitud.
- Enum: la declaración de un tipo enumerado es simplemente una lista de valores que pueden ser tomados por una variable de este tipo y se presenta con identificadores.
- Float: se utiliza un dato en coma flotante de 32 bits, almacena valores con una precisión aproximada de 7 dígitos; se añade al final del valor la letra f para declarar que es del tipo float.
- Double: se utiliza para declarar un dato en coma flotante de 64 bits, almacena valores con una precisión aproximada de 16 dígitos. Por defecto una constante es considerada del tipo double.

Tipos derivados

Los tipos derivados son construidos a partir de los tipos primitivos. Algunos de ellos son: estructuras, uniones, matrices (arrays), punteros y funciones.

Punteros

Cada variable de un programa tiene una dirección en la memoria del ordenador. Esta dirección indica la posición del primer byte que la variable ocupa. En el caso

de una estructura es la dirección del primer campo. Como en cualquier caso las variables son almacenadas ordenadamente y de una forma predecible, es posible acceder a estas y manipularlas mediante otras variables que contenga su dirección. A este tipo de variables se les denomina punteros.

Los punteros C son el tipo más potente y seguramente la otra clave del éxito del lenguaje. La primera ventaja que obtenemos de los punteros es la posibilidad que nos dan de poder tratar con datos de un tamaño arbitrario sin tener que moverlos por la memoria. Esto puede ahorrar un tiempo de computación muy importante en algunos tipos de aplicaciones. También permiten que una función reciba y cambie el valor de una variable. Mediante un puntero a una variable podemos modificarla indirectamente desde una función cualquiera.

Un puntero se declara de la forma: tipo *nombre;

Para manipular un puntero, como variable que es, se utiliza su nombre; pero para acceder a la variable a la que apunta se le debe preceder de *. A este proceso se le llama indirección. Para trabajar con punteros existe un operador, &, que indica 'dirección de'. Con él se puede asignar a un puntero la dirección de una variable, o pasar como parámetro a una función.

Un puntero iniciado correctamente siempre apunta a un objeto de un tipo particular, un puntero no iniciado no se sabe a dónde apunta.

Estructuras

Una estructura es un conjunto de datos agrupados que forman una entidad lógica que mantengan algún tipo de relación, aunque sean de tipos diferentes; se referencian bajo un mismo nombre, para hacer más eficiente su manejo.

Las estructuras ayudan a organizar datos complicados, particularmente en programas grandes, ya que permiten tratar como una unidad a un conjunto de variables relacionadas, en lugar de tratarlas como entidades independientes.

Un buen criterio de uso de estructuras establece que la definición de una estructura debe ir acompañada por la definición de un conjunto de funciones que se encargan de realizar la manipulación directa de la nueva entidad creada.

Una estructura se define en lenguaje C a través de la siguiente sintaxis:

```
struct [<identificador>
{
    [<tipo><nombre_objeto>[,<nombre_objeto>,...]];
} [<objeto_estructura>[,<objeto_estructura>,...];
```

El identificador de la estructura es un nombre opcional para referirse a la estructura. Los objetos de estructura son objetos declarados del tipo de la estructura, y su inclusión también es opcional. En el interior de una estructura, entre las llaves, se pueden definir todos los elementos que consideremos necesarios, del mismo modo que se declaran los objetos.

La declaración:

```
struct Nombre Var1;
{
    tipo1 Campo1;
    tipo2 Campo2;
    ...
    tipoNCampoN;
};
```

Declara una variable del tipo "struct Nombre", esto es, el compilador reserva la cantidad de memoria suficiente para mantener la estructura íntegra (es decir espacio para almacenar Campo1, Campo2,..., CampoN). Cuando se hace referencia a la variable Var, se está haciendo referencia a la estructura íntegra.

Funciones en el interior de estructuras

C++, permite incluir funciones en el interior de las estructuras. Normalmente estas funciones tienen la misión de manipular los datos incluidos en la estructura, y su uso está muy relacionado con la programación orientada a objetos.

Dos funciones muy particulares son las de inicialización, o constructor, y el destructor. El constructor es una función sin tipo de retorno y con el mismo nombre que la estructura (usando constructores nos aseguramos los valores iniciales para los elementos de la estructura). El destructor tiene la misma forma, salvo que el nombre va precedido el símbolo "~".

Por supuesto se pueden definir otras funciones y también constructores más elaborados e incluso, redefinir operadores. Y en general, las estructuras admiten cualquiera de las características de las clases, siendo en muchos aspectos equivalentes.

La asignación de estructuras está permitida, pero sólo entre objetos del mismo tipo de estructura, (salvo que se usen constructores), y funciona como la intuición nos dice que debe hacerlo.

Podemos establecer dos tipos de estructuras diferentes: estáticas y dinámicas, de las cuales podemos decir que sirven para registrar conjuntos de valores.

Estructuras estáticas

Dentro de las estructuras estáticas tenemos las matrices que son la forma de registrar conjuntos de valores todos del mismo tipo o relacionados entre sí y almacenados consecutivamente en memoria, por ejemplo: almacenar los datos relativos a una persona como su nombre, dirección, teléfono, etc.

En general, la representación de matrices se hace mediante variables suscritas o de subíndices.

Es necesario definir una matriz antes de utilizarla y se especifica el nombre, el número de elementos de la misma y el tipo de estos.

Estructuras dinámicas

La principal característica que tienen las estructuras dinámicas es la facultad que tienen para variar su tamaño y hay muchas situaciones que requieren este tipo de estructura

No es posible asignar una cantidad fija de memoria para una estructura dinámica, y como consecuencia un compilador no puede asociar direcciones explícitas con los elementos de tales estructuras. Se resuelve este problema realizando asignación dinámica para los elementos individuales, al tiempo que son creados durante la ejecución del programa. Podemos destacar las siguientes estructuras:

- Pilas: es una lista lineal en la que todas las intersecciones y supresiones se hacen en un extremo de la lista.
- Colas: es una lista lineal en la que todas las intersecciones se hacen por el extremo final y todas las supresiones se hacen por el extremo inicial.
- Árboles: es una estructura no lineal formada por un conjunto de nodos y un conjunto de ramas. En este existe un nodo especial denominado raíz, un nodo del que sale alguna rama llamado nodo de bifurcación o nodo rama y un nodo que no tiene rama recibe el nombre de nodo terminal o nodo hoja.

Funciones

Una función es un conjunto de instrucciones que realizan una tarea específica y al que nos referimos mediante un nombre. En un programa C se distinguen dos clases de funciones, las funciones definidas por el usuario y las funciones de bibliotecas.

Declaración de una función

La declaración de una función, también conocida como prototipo de la función, indica, además del nombre de la función, cuantos parámetros y que tipo tienen, así como el tipo del valor retornable; para evitar errores durante la compilación. Para indicar que una función no retorna nada se utiliza la palabra reservada void.

Una función puede ser declarada implícitamente o explícitamente.

- La declaración implícita se da cuando la función es llamada y no existe una declaración previa, ejemplo la función main es declarada implícitamente.
- La declaración explícita se recomienda en C++ para el tipo de los parámetros de la función, así como el tipo del valor retornado.

Cuando una función definida en el programa se invoca a una función de biblioteca de C es necesario añadir su prototipo sin embargo las declaraciones de las funciones pertenecientes a las bibliotecas estándar de C son proporcionadas por los ficheros de cabecera o ficheros .h y se hace al especificar la sintaxis de las funciones de biblioteca y el fichero de cabecera donde está declarado.

Definición de una función

La definición de una función consta de una cabecera de función, del cuerpo de la función encerrada entre llaves ({}).

Las variables declaradas en el cuerpo de la función son locales y por definición solamente son accesibles dentro del mismo. El tipo del resultado especifica el tipo de los datos retornados por la función y puede ser del tipo primitivo o derivado, pero no puede ser una matriz o una función, por defecto se supone int si este no se especifica.

La lista de parámetros de una función está compuesta por las variables que reciben los valores especificados cuando es invocada. Consiste de una lista de cero, uno o más identificadores con sus tipos, separados por coma.

Llamada a una función

Llamar una función es sinónimo de ejecutarla, se hace desde otra función o incluso de ella misma. Dicha llamada está formada por el nombre de la función seguida de una lista de argumentos, denominados también parámetros actuales.

4.3.3. ORIENTACIÓN A OBJETOS

La programación orientada a objetos (también conocida como O.O.P., por sus siglas en inglés) es lo que se conoce como un paradigma o modelo de programación que utiliza objetos y sus interacciones, para diseñar aplicaciones y programas. Esto significa que no es un lenguaje específico, o una tecnología, sino una forma de programar, una manera de plantearse la programación. No es la única (o necesariamente mejor o peor que otras), pero se ha constituido en una de las formas de programar más populares e incluso muchos de los lenguajes que usamos hoy día lo soportan o están diseñados bajo ese modelo.

Lo que caracteriza a la POO es que intenta llevar al mundo del código lo mismo que encontramos en el mundo real.

A continuación presentamos una serie de definiciones importantes en la POO:

Definición (objeto)

Entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos) los mismos que consecuentemente reaccionan a eventos.

Definición (clase)

Es un tipo definido por el usuario que describe los atributos y los métodos de los objetos que se crearan a partir de la misma. Una clase es como la definición de un objeto, pero no es el objeto en sí.

Definición (atributo)

Constituye la estructura interna de los objetos de una clase. En C++ un atributo también se denomina dato miembro

Definición (método)

Los métodos generalmente forman lo que se denominan interface o medio de acceso a la estructura interna de los objetos; ellos definen las operaciones que se pueden realizar con sus atributos. En C++ un método de una clase también se denomina función miembro de la clase.

Definición (mensaje)

Una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó.

Definición (evento)

Es un suceso en el sistema. El sistema maneja el evento enviando el mensaje adecuado al objeto pertinente. También se puede definir como evento, a la reacción que puede desencadenar un objeto, es decir la acción que genera.

Características de la POO

En la programación orientada a objetos se consideran las siguientes características:

Definición (abstracción):

Denota las características esenciales de un objeto, donde se capturan sus comportamientos. Cada objeto en el sistema sirve como modelo de un "agente" abstracto que puede realizar trabajo, informar y cambiar su estado, y "comunicarse" con otros objetos en el sistema sin revelar cómo se implementan estas características. La abstracción es clave en el proceso de análisis y diseño orientado a objetos, ya que mediante ella podemos llegar a armar un conjunto de clases que permitan modelar la realidad o el problema que se quiere atacar.

La abstracción nos permite dividir nuestro programa en distintos objetos que se agrupan para formar cosas más complejas, básicamente es la capacidad de separar los elementos (al menos mentalmente) para poder verlos de forma singular.

Definición (encapsulación)

También conocida como ocultamiento. La encapsulación se encarga de mantener ocultos los procesos internos que necesita para hacer lo que sea que haga, dándole al programador acceso sólo a lo que necesita. Esto da dos ventajas iniciales: Lo que hace el usuario puede ser controlado internamente (incluso sus errores), evitando que todo colapse por una intervención indeseada. La segunda ventaja es que, al hacer que la mayor parte del código esté oculto, se puede hacer cambios y/o mejoras sin que eso afecte el modo como los usuarios van a utilizar el código. Sólo se tiene que mantener igual la forma de acceder a él, estas puertas de acceso que se dan a los usuarios son lo que se conoce como interfaz.

Definición (herencia)

Uno de los elementos más interesantes de la P.O.O. La herencia es la capacidad que tiene una clase de derivar las propiedades y métodos de otra.

La herencia nos permite, entre otras cosas, evitar tener que escribir el mismo código una y otra vez, puesto que al definir que una categoría (clase) pertenece a otra, automáticamente estamos atribuyéndoles las características generales de la primera, sin tener que definir las de nuevo.

Definición (polimorfismo)

Refiere que comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre, al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando.

Bibliotecas básicas

Una biblioteca de C es una colección de funciones utilizadas en el lenguaje de programación C. Las bibliotecas más comunes son la biblioteca estándar de C

La biblioteca estándar de C es una recopilación de ficheros cabecera y bibliotecas con rutinas, que implementan operaciones comunes, tales como las de entrada y salida o el manejo de cadenas. C no incluye palabras clave para estas tareas, por lo que prácticamente todo programa implementado en C se basa en la biblioteca estándar para funcionar. Algunas de las bibliotecas usadas en nuestro trabajo son

Vector

Está definida en el espacio de nombres std facilita la creación y manipulación de matrices.

String

Contiene la declaración de una colección de funciones útiles para manejar cadenas y otros array de caracteres (proporciona métodos para examinar caracteres individuales)

Map

Está definida en el espacio de nombres std facilita la creación y manipulación de matrices asociativas.

ctime

Se especializa en manipulación de fechas y horas.

assert.h

Contiene la macro assert (aserción), utilizada para detectar errores lógicos y otros tipos de fallos en la depuración de un programa (imprime un mensaje de error y aborta el programa).

iostream

Proporciona la funcionalidad necesaria para acceder secuencial o aleatoriamente a un fichero.

fstream

Se especializa en manipular ficheros para abrir y cerrar archivos, así como para realizar las operaciones de lectura y escritura de datos en archivos.

cmath

Contiene la declaración de algunas funciones para realizar operaciones matemáticas comunes sobre valores de tipo "double".

stdio

Incluye macros y funciones para realizar operaciones de entrada y salida sobre ficheros y flujos de datos.

Cstdlib

Contiene las declaraciones de una colección de funciones útiles y la definición de tipos y macros para usarlas.

4.4. TEORÍA DE COLAS

Una herramienta fundamental para la validación de nuestro trabajo está en un tema estudiado mucho por las matemáticas llamado Teoría de Colas.

Las colas son un aspecto de la vida que nos encontramos continuamente en nuestras actividades. Los embotellamientos en los semáforos, las filas de espera en el súper mercado, en los cajeros automáticos, al abordar un autobús, etc., este fenómeno surge cuando unos recursos compartidos necesitan ser accedidos para dar servicio a un elevado número de trabajos o clientes. Dicho en otras palabras se da cuando el número de clientes deseando usar un servicio es mucha y el número de servidores es poco.

Para ayudar un poco a comprender sobre este tema daremos a continuación algunas definiciones básicas:

Definición (fuente de entrada o población potencial):

Es un conjunto de individuos (no necesariamente seres vivos) que pueden llegar a solicitar el servicio en cuestión. Podemos considerarla finita o infinita

Definición (cliente):

Es todo individuo de la población que solicita servicio. Suponiendo que los tiempos de llegada de clientes consecutivos son $t_0 < t_1 < t_2 < \dots$, será importante conocer el patrón de probabilidad según el cual llegan los clientes.

Definición (cola):

Propiamente dicha, es el conjunto de clientes que hacen espera, es decir los clientes que ya han solicitado el servicio pero que aún no han pasado al mecanismo de servicio.

Definición (capacidad de la cola):

Es el máximo número de clientes que pueden estar haciendo cola. De nuevo, puede suponerse finita o infinita.

Definición (disciplina de la cola):

Es el modo en el que los clientes son seleccionados para ser servidos. Las disciplinas más habituales son:

- La disciplina FIFO (first in first out), también llamada FCFS (first come first served): según la cual se atiende primero al cliente que antes haya llegado.
- La disciplina LIFO (last in first out), también conocida como LCFS (last come first served) o pila: que consiste en atender primero al cliente que ha llegado de último.
- La RSS (random selection of service), o SIRO (service in random order), que selecciona a los clientes de forma aleatoria.

Definición (mecanismo de servicio):

Es el procedimiento por el cual se da servicio a los clientes que lo solicitan. Para determinar totalmente el mecanismo de servicio debemos conocer el número de servidores de dicho mecanismo. Del servicio de colas se debe especificar la distribución de probabilidad de los tiempos de servicio. La distribución más usada para los tiempos de servicio es la exponencial.

Definición (sistema de cola):

Es el conjunto formado por la cola y el mecanismo de servicio, junto con la disciplina de la cola.

4.4.1. MEDIDAS DE RENDIMIENTO DE UN SISTEMA DE COLAS

La siguiente notación muestra la forma en que se puede evaluar la eficiencia de un sistema de cola.

P_n : Probabilidad de que haya exactamente n clientes en el sistema.

L_s : Número esperado de clientes en el sistema.

L_q : Número esperado de clientes en la cola.

W_s : Tiempo de espera en el sistema para cada cliente.

W_q : Tiempo de espera en la cola para cada cliente.

ρ : Un factor de utilización del sistema.

Debido a las múltiples variantes que presenta un sistema de cola fue necesario clasificarlos según sus características.

A continuación hablaremos de los modelos más utilizados en la teoría de colas nombrados según para ello usaremos la notación que introdujo David G. Kendall en 1953.

4.4.2. CLASIFICACIÓN DE LOS MODELOS DE COLA SEGÚN KENDALL

David G. Kendall estableció en 1953 una notación para los sistemas de colas que hace referencia a su proceso de llegada, su tiempo de servicio, la cantidad de sus

servidores y su capacidad máxima, quedando entonces caracterizado un sistema de colas mediante una 4-upla de la forma $W/X/Y/Z$.

El proceso de llegada (W) puede ser referido mediante uno de los siguientes símbolos, los cuales indican cómo se distribuye el tiempo entre llegada de los clientes al sistema:

M: Exponencial con tasa de llegada λ

D: Determinista

G: General

El tiempo de servicio (X) puede ser referido mediante uno de los siguientes símbolos, los cuales indican cómo se distribuye el tiempo que tarda un cliente en ser atendido:

M: Exponencial con una velocidad de servicio μ

D: Determinista

G: General

Servidores (Y)

Hace referencia al número de servidores que hay en el sistema, este puede tomar los valores de 1 si el servidor es único o "c" si la cantidad de servidores es mayor que uno.

La capacidad máxima (Z), hace referencia al número máximo de clientes que puede haber en el sistema, si la cantidad de clientes es ilimitada, no se especifica.

Por ejemplo:

El sistema de cola $M/M/1/K$ corresponde a un sistema en el que los tiempos entre llegadas están distribuidos exponencialmente (M); los tiempos de servicio están exponencialmente distribuidos (M); hay un único servidor (1); y solo se permiten K clientes en el sistema.

Ahora pasaremos a mostrar algunos modelos y sus respectivas medidas de rendimientos.

4.4.3. MODELOS DE COLAS MÁS COMUNES

Modelo $M/M/1$

La tasa de llegada es exponencial, tiempos de servicio exponencial, disciplina FIFO, un solo servidor y población infinita.

Sus correspondientes medidas de rendimiento se calculan mediante las siguientes fórmulas:

$$L_s = \frac{\lambda}{\mu - \lambda} \qquad L_q = \frac{\lambda^2}{\mu(\mu - \lambda)} = \rho * L_s$$

$$W_s = \frac{1}{\mu - \lambda} \qquad W_q = \frac{\lambda}{\mu(\mu - \lambda)} = \rho * W_s$$

$$P_o = 1 - \frac{\lambda}{\mu} = (1 - \rho) \qquad P_n = \left(1 - \frac{\lambda}{\mu}\right) * \left(\frac{\lambda}{\mu}\right)^n = (1 - \rho) * \rho^n$$

$$P_{n>k} = \left(\frac{\lambda}{\mu}\right)^{k+1}$$

$P_{n>k}$ = Probabilidad de que más de "k" individuos estén en el sistema.

Modelo M/M/c

La tasa de llegada es exponencial, tiempos de servicio siguen una distribución exponencial, disciplina FIFO, el número de servidores es mayor que 1 y población infinita.

Sus medidas de rendimiento son las siguientes:

c = número de servidores en uso.

$$P_o = \frac{1}{\left[\sum_{n=0}^{c-1} \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n \right] + \frac{1}{c!} \left(\frac{\lambda}{\mu}\right)^c \frac{c\mu}{c\mu - \lambda}} \text{ para } c\mu > \lambda$$

$$L_s = \frac{\lambda \mu \left(\frac{\lambda}{\mu}\right)^c}{(c-1)! (c\mu - \lambda)^2} P_o + \frac{\lambda}{\mu}$$

$$L_q = L_s - \frac{\lambda}{\mu} = L_s - \rho$$

$$W_s = \frac{\mu \left(\frac{\lambda}{\mu} \right)^c}{(c-1)!(c\mu - \lambda)^2} P_0 + \frac{1}{\mu} = \frac{L_s}{\lambda}$$

$$W_q = W_s - \frac{1}{\mu} = \frac{L_q}{\lambda}$$

Modelo M/G/1

La tasa de llegada es exponencial, tiempos de servicio siguen una distribución General, disciplina FIFO, y una población infinita.

Sus correspondientes medidas de rendimiento son:

$$L_s = L_q + \rho \qquad L_q = \frac{\lambda^2 \sigma^2 + \rho^2}{2(1-\rho)}$$

$$W_s = W_q + \frac{1}{\mu} \qquad W_q = \frac{L_q}{\lambda}$$

$$P_0 = 1 - \rho \qquad P_n = \rho$$

Los modelos M/M/1 son útiles para conocer la influencia de los distintos parámetros en los sistemas básicos mientras que modelos M/G/1 permiten considerar una clase más general de sistemas que comparten recursos.

5. LA SIMULACIÓN DE TRÁFICO

A continuación presentamos características del tráfico vehicular en el centro de la ciudad de León, tomando como base de datos de entrada, los datos provenientes del trabajo de campo, en el que se realiza una serie de observaciones en las calles del sistema vial tomado.

5.1. CONSTRUCCIÓN DE LA RED

La red de tráfico en toda su estructura se modela mediante un grafo cuyos nodos representan las esquinas de las vías y los arcos del grafo representan las vías de la red.

Para una mejor comprensión, explicamos algunos términos referidos en el trabajo:

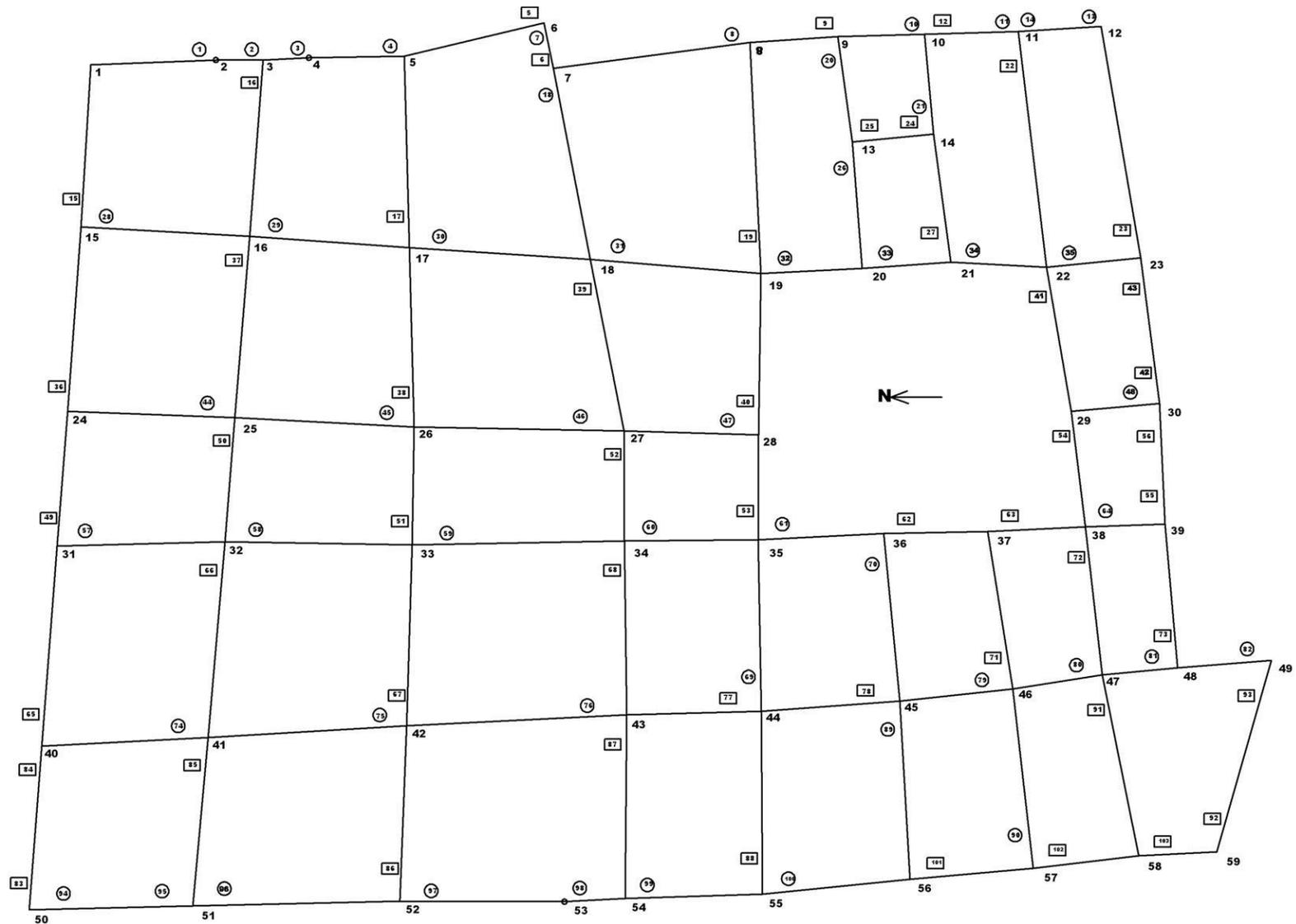
Vías: Es tomado como el trayecto que une dos esquinas o puntos llamados nodos, es decir es una recta que une a dos puntos distintos, donde los objetos (vehículos) se trasladan desde el punto inicial de la vía hasta el punto final de la vía.

Nodos: Es un punto $P(x, y)$, el cual representa una esquina de la red de tráfico.

Giros: Es un par ordenado de vías, la primera de las cuales es la vía por donde vienen los vehículos y la segunda es la vía hacia donde se dirigen los vehículos.

Semáforo: está concebido como un aparato que deja pasar e impide el tráfico en determinados giros durante períodos establecidos de tiempo llamados fases. El tiempo que toma un cambio en la configuración de giros permitidos y prohibidos se llama tiempo de fase. En la simulación el semáforo se caracteriza por una lista de giros no permitidos asociados con cada fase del semáforo. Se considera que el semáforo solo tiene dos colores rojo y verde.

La red considerada para la simulación incluye las vías que inician una cuadra al norte de la esquina noroeste del Parque San Juan (nodo 1 en la red de la figura 1), desplazándose hacia el oeste hasta la esquina ubicada 2 cuadras al oeste de la Iglesia San José (nodo 50 de la figura 1), luego la frontera de la red se extiende desde esa esquina hacia el sur hasta la esquina oeste del llamado Callejón de la Renta (nodo 59 de la figura 1), de esa esquina hasta la esquina este de ese mismo callejón (nodo 49 de la figura 1), luego hacia el norte hasta la esquina oeste del Callejón del Teatro (nodo 48 de la figura 1), luego hacia el este hasta llegar a 1 cuadra al este de la esquina sureste del HEODRA (nodo 12 de la figura 1), doblando hacia el norte hasta llegar al Tope Gurdían (nodo 7 de la figura 1), luego hacia el este hasta llegar al Tope Mollieri (nodo 6 de la figura 1), y finalmente hacia el norte, hasta llegar a la esquina inicial (nodo 1 de la figura 1).



RED DE TRAFICO VEHICULAR DEL CENTRO DE LEON, NICARAGUA

- - IDENTIFICADOR DE VIA SIN ALTO
- - IDENTIFICADOR DE VIA CON ALTO

5.2. EL TRABAJO DE CAMPO

Mediante georeferenciación se obtuvieron las coordenadas geográficas UTM de los nodos (intersecciones o esquinas) del área delimitada. Las vías reales de la red de tráfico son sustituidas por segmentos de rectas que unen pares de nodos conectados por las vías de la red real. La red incluye 59 nodos y 103 vías diferentes.

Con la información obtenida se construyeron dos archivos de datos uno que contiene el identificador de cada nodo y sus coordenadas UTM, el otro que contiene identificadores de vías y los nodos inicial y final de cada vía. Estos archivos se utilizan para alimentar el programa C++ de nuestro modelo de tal forma que podemos obtener la red de tráfico requerida en un plano cartesiano.

Seguidamente se identificaron las señales de tránsito (semáforos, altos, giros, etc.) en cada una de las vías que comprende la red para conocer la fluidez y el comportamiento de los vehículos en la vía de tal forma que el modelo se ajuste al sistema real.

Para caracterizar el comportamiento de los vehículos en la red se realizó la recolección de la frecuencia de giros de los vehículos durante periodos de tiempos considerados como horas picos los cuales son:

6:50am-8:10am
9:00am-10:00am
11:40am-1:10pm
3:00pm-4:00pm
5:50pm-6:10pm

Dichos periodos fueron divididos en sub-períodos de 20 minutos, esto para tener una mejor medida del comportamiento del tráfico, la toma de datos se realizó con la utilización de un formato para cada nodo de la red (ver anexo 9.1.).

Con los datos obtenidos se conformó una base de datos completa de todos los nodos en una hoja de Excel y en la cual se especifica:

- Las vías de inicio y finalización del giro
- Frecuencia de giro por período de tiempo

En la simulación los vehículos que circulan por la red de tráfico realizan giros en la red con una frecuencia que se toma de la base de datos anterior.

5.3. MEDIDAS DE DESEMPEÑO DE LA RED DE TRÁFICO

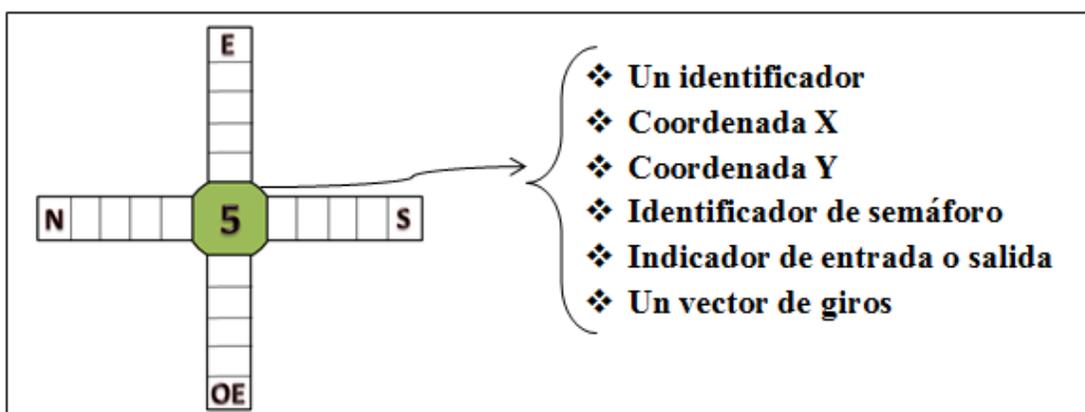
Para valorar la eficiencia de la red de tráfico del sistema real tomamos las siguientes medidas de rendimiento:

- Cantidad promedio de vehículos en cada vía, esto es el número promedio de vehículos que están en la vía en cualquier instante.
- Tiempo promedio que tarda un vehículo en cada vía, esto es el tiempo promedio que tarda un vehículo en transitar desde el nodo inicial hasta el nodo final de la vía.
- Probabilidad de encontrar n vehículos en la vía en un instante cualquiera.
- Tiempo promedio que tardan los vehículos en cruzar la intersección.

5.4. CARACTERIZACIÓN DE LOS ELEMENTOS DE LA SIMULACIÓN

Las más importantes conceptualizaciones usadas para modelar las componentes de la red de tráfico son:

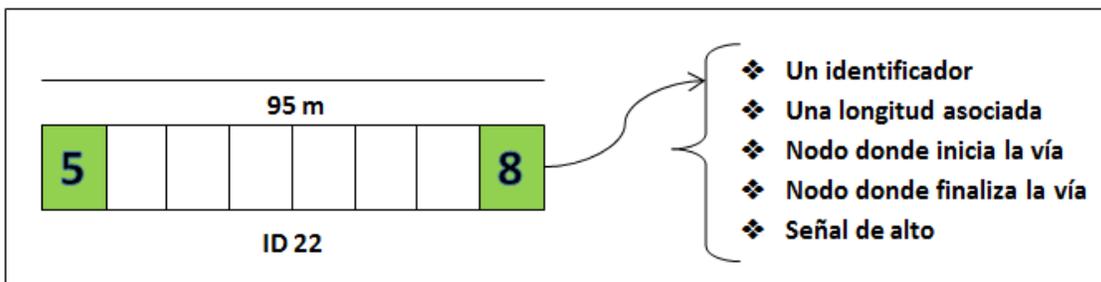
- a) Las intersecciones en la red de tráfico o nodos de la red, son objetos formados por:
 - Un identificador, es decir un nombre o número de nodo.
 - La coordenada x y la coordenada y de la intersección las cuales sirven para determinar la ubicación de los nodos en la red.
 - Un indicador de semáforo, el cual establece si hay en esa intersección un semáforo. Hay que hacer notar que el indicador de semáforo es un puntero a un objeto de tipo semáforo como será descrito posteriormente.
 - La característica que identifica al nodo como un nodo por la cual pueden entrar vehículos externos a la red, o como un nodo por el cual pueden salir vehículos de la red.
 - Un vector de giros que almacena la información acerca de los giros que puede realizar un vehículo cualquiera al pasar por esa intersección.



Esquema de la Caracterización de un nodo

b) Las vías en la simulación se dividieron en celdas de 5.5 ms, de tal forma que en un momento dado una celda está ocupada, si hay un vehículo en esa posición o vacía si no hay vehículo en ella. Las vías son modeladas mediante objetos con los siguientes atributos:

- Un identificador, es decir un número de vía.
- Una longitud asociada.
- El nodo desde donde parte y el nodo hacia donde llega la vía.
- En el caso de calles con doble vía estas son sustituidas por dos vías separadas, lo cual presupone que no hay interacción entre los tráficos circulando sobre la misma calle física. Esto constituye una simplificación de la realidad que no causa mayores alteraciones a la simulación.
- Una señal de alto define de manera permanente la prioridad de circulación en las vías afectadas por el. En la simulación es un atributo del objeto vía definido como una variable booleana que indica si al final de la vía esta posee una señal de alto o no.



Esquema de la Caracterización de una Vía

c) Un giro se concibe como una estructura que describe cuales son las vías de llegada y salida de la intersección de un vehículo cualquiera que llega a una intersección y la cruza. Formalmente está constituida por dos componentes punteros que apuntan a dos vías, la vía inicial y la vía final del giro.

d) El semáforo está concebido como un aparato que cambia de manera temporal el régimen de prioridad de circulación en las vías afectadas por el. En la simulación un semáforo es un objeto con los siguientes atributos:

- Un identificador.
- El identificador del nodo de la red en la que se ubica.

- Un tiempo de fase, que indica cada cuanto tiempo el semáforo cambia la prioridad de circulación en las vías. Esto después de haber observado que en el centro de la ciudad estos cambian en intervalos de tiempo constantes.
- Numero de fases, que establece cuantos posibles esquemas de prioridad de circulación tiene el semáforo.
- Prohibiciones del semáforo, es un vector que contiene los giros que no pueden ser realizados en cada una de las fases del semáforo.
- Los vehículos usados actualmente en la simulación son objetos sencillos los que tienen como atributos un identificador y una velocidad limitada a una celda por unidad de tiempo. Versiones más sofisticadas, no consideradas en este trabajo, podrían incluir reglas para acelerar o desacelerar en dependencia del resto de conductores.

5.4.1. CIRCULACIÓN DE LOS VEHÍCULOS EN LA RED DE TRÁFICO

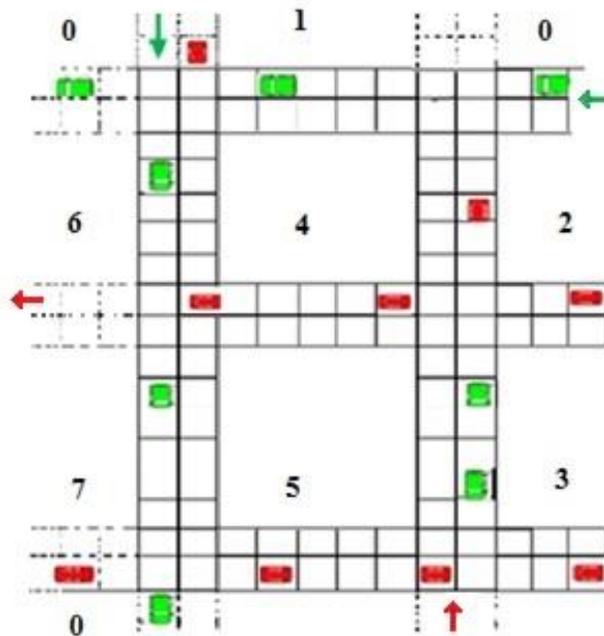
La simulación se da en tiempo discreto. En cada momento se parte de la posición de los vehículos en la red, se usan determinadas reglas para actualizar la posición de los vehículos en la red.

Inicialmente se mueven los vehículos partiendo del inicio de la vía hasta la parte frontal de la misma, posteriormente se pasa al movimiento de los vehículos en las intersecciones.

En la simulación inicialmente se ubica un vehículo al inicio de cada una de las vías, y este usa la forma más simple de desplazamiento, el avanza a una velocidad de una casilla por segundo (aproximadamente 20 km/h) en el caso que la celda que le precede este libre, en caso contrario el no avanza en el transcurso de ese tiempo.

5.4.2. INGRESO Y SALIDA DE LA RED DE TRÁFICO

La única forma de ingreso a la red considerada es a través de los nodos de ingreso, en los que la simulación define giros posibles que inician en la vía ficticia cero. La salida igualmente se produce como un giro que se realiza en un nodo caracterizado como de salida, y él se produce como giro desde una vía de entrada al nodo hacia la vía ficticia cero.



Esquema de la circulación, entrada y salida de los vehículos en la red de tráfico vehicular

5.4.3. LOS SEMÁFOROS

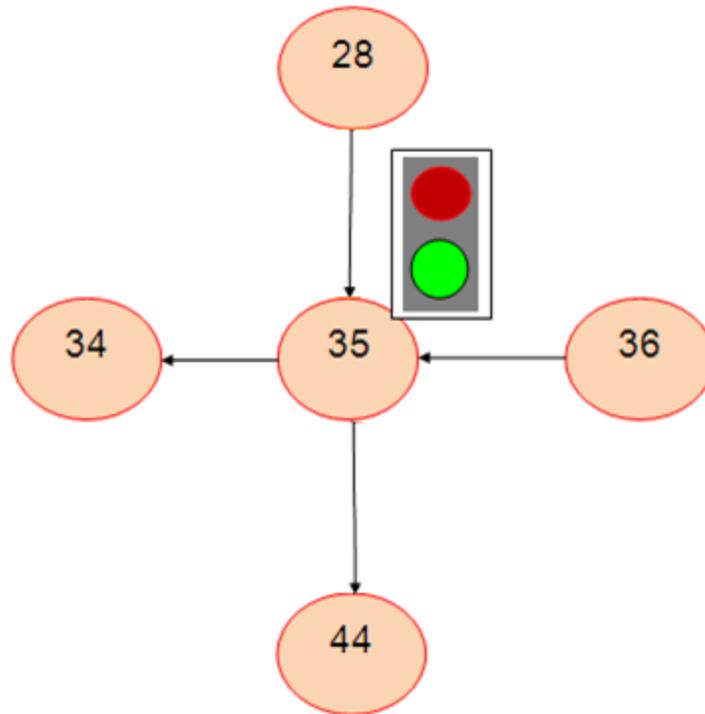
Los semáforos son parte importante dentro del sistema del flujo vehicular ya que una adecuada programación o configuración de los semáforos ayuda a conseguir un flujo relativamente libre y mantiene los tiempos de espera de los conductores en niveles aceptables, mientras que una inadecuada configuración produce congestiones y tiempos de espera elevados.

El funcionamiento eficiente de los semáforos requiere de una adecuada temporización de las diferentes luces del semáforo. No es un objetivo del proyecto crear modelos o realizar optimizaciones en la programación de los semáforos.

Para la simulación, el semáforo se considera como un tipo especial de señal de tránsito, se le asignaron características de acuerdo al del sistema real, es decir un determinado tiempo de fase medido en segundos y dos fases una verde y otra roja.

En la simulación cuando un vehículo llega al final de la vía, si el semáforo está en verde y la vía de destino no está ocupada el realiza el cruce, en caso contrario, el espera; si el semáforo está en rojo el deberá esperar el tiempo de cambio de fase. Los cruces en los semáforos se ajustaran de acuerdo a la probabilidad asignada por la frecuencia de giros.

En la red de tráfico estudiada solo encontramos un semáforo ubicado en la esquina de la Casa CUUN. Cuya caracterización es la siguiente:



Nodos:

Nodos	Posición
36	Esquina Sur Casa CUUN
34	Esquina Norte Casa CUUN
35	Esquina Casa CUUN
28	Esquina Este Casa CUUN
44	Esquina Oeste Casa CUUN

Vías

Vías de entrada	Vías de salida
Vía 52: va del nodo 28 al nodo 35	Vía 59: va del nodo 35 al nodo 34
Vía 60: va del nodo 36 al nodo 35	Vía 68: va del nodo 35 al nodo 44

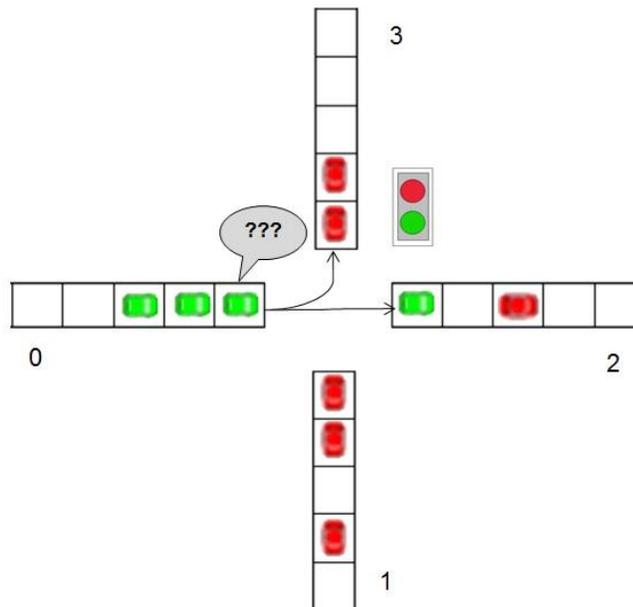
Giros

Posibles giros de la vía 52	Posibles giros de la vía 60
Vía 52-Vía 59	Vía 60-Vía 59
Vía 52-Vía 68	Vía 60-Vía 68

5.4.4. EL CRUCE DE LAS INTERSECCIONES

Para realizar el cruce de las intersecciones los vehículos siguen las siguientes reglas:

- Para cada vía de entrada a la intersección
 - Si la vía de entrada tiene un vehículo adelante
 - Se revisan restricciones de pase por señales de tráfico
 - Se determina un giro a realizar conforme a la frecuencias de giro del nodo
 - Si su vía de destino está desocupada, se da el giro
- Si no fue posible hacer un giro se examina la siguiente vía de entrada



Esquema de la caracterización del cruce de un nodo con un semáforo.

5.4.5. CLASES Y FUNCIONES EN LA SIMULACIÓN

En las siguientes tablas presentamos una caracterización de todos los datos variables, clases y funciones del lenguaje de programación C++ utilizadas en la construcción del programa para el modelo, de tal forma que se describe una clasificación y la utilidad que tiene cada clase como estructura fundamental.

Constantes del programa

N	T	V	D
TF	Int	3600	Momento en que finaliza la simulación
NP	Int	5	Cantidad de períodos de tiempo que cubre la simulación
LP	Int	8640	Tamaño de cada período de tiempo en la simulación
TA	Int	1	Longitud del intervalo de tiempo en que se da la simulación
LC	double	01/05/50	Longitud en mts. de las celdas de cada vía
PI	double	3.14159	Número PI
MV	Int	10	Máxima capacidad de vehículos en las vías de la red
TI	Int	2	Momento inicial de la simulación

Clases y/o estructuras en el programa

N	D
Veh	Vehículo
Sem	Semáforo
Prsem	Bloqueo de semáforo
Nodo	Nodo de la red de tráfico
Via	Vía de la red de tráfico

Tipos en el programa

N	T	D
Tie	Int	variables asociadas a tiempos
Ide	Long	variables asociadas a identificadores
Cor	Double	variables asociadas a coordenadas
Vel	Double	variables asociadas a velocidades
Lon	Double	variables asociadas a longitudes
Pro	Double	variables asociadas a probabilidades
Tas	Double	variables asociadas a tasas
Vco	vector<Cor>	vector con cada componente asociadas a una coordenada
Cel	vector<Veh*>	vector con cada componente apuntando a un vehículo
PrS	vector<Prsem*>	vector con cada componente apuntando a un bloqueo
VSe	vector<Sem*>	Vector con cada componente apuntando a un semáforo
Nodos	map<Ide,Nodo*>	Aplicación de punteros a nodos en función de su identificador
Vias	map<Ide,Vias*>	Aplicación de punteros a vías en función de su identificador

VARIABLES GLOBALES EN EL PROGRAMA

Pe	Tie	Período de tiempo actual en la simulación
Tg	Tie	Tiempo actual en la simulación
Cv	Int	Cantidad de vehículos que han entrado a la simulación
Tiemvia	Ofstream	Nombre archivo para tiempos promedios de recorrido de las vías
vehvia	Ofstream	Nombre archivo para cantidades promedios de vehículos en las vías
Probest	Ofstream	Nombre archivo para probabilidades de hallar una cantidad dada de vehículos en las vías
Tievia	Ofstream	Nombre archivo para tiempos promedios adelante de vehículos en las vías
Giros	Vgi	Vector que guarda todos los giros de la red
Vsem	Vse	Vector de punteros a los semáforos de la red
Nodos	Nodos	Aplicación que contiene punteros a nodos como función de su identificador
Vías	Vias	Aplicación que contiene punteros a vías como función de su identificador

En el siguiente apartado se explican los comandos aplicados para la estructuración de las clases.

La clase Veh (vehículo)

Tiene como atributos

N	T	D
id_	Ide	Guarda el identificador del vehículo
tvia_	Tie	Guarda el tiempo que el vehículo lleva sobre la vía actual
tavia_	Tie	Guarda el tiempo que el vehículo lleva adelante sobre la vía actual

Tiene como métodos

N	T	TA	D
f_id	Void	Ide	Asigna identificador al vehículo
Id	Ide	--	Da el identificador del vehículo
f_tvía	Void	Tie	Asigna valor al tiempo que el vehículo lleva sobre la vía actual
i_tvía	Void	--	Pone en cero el tiempo que el vehículo lleva sobre la vía actual
Tvía	Tie	--	Da el tiempo que el vehículo lleva sobre la vía actual
f_tavía	Void	Tie	Asigna valor al tiempo adelante sobre la vía actual del vehículo
i_tavía	Void	--	Pone en el tiempo adelante sobre la vía actual del vehículo
Tavía	Tie	--	Da el tiempo que el vehículo lleva adelante sobre la vía actual

La clase Sem (semáforo)

Tiene como atributos

N	T	D
id_		Guarda el identificador del semáforo
tf_	Tie	Guarda el tiempo de fase del semáforo
nf_	Int	Guarda el número de fases del semáforo
Prh_	PrS	Guarda la lista de punteros a vías bloqueadas según la fase

Tiene como métodos

N	T	TA	D
f_id	Void	Ide	Asigna un identificador al semáforo
Id	Ide		Da el identificador del semáforo
f_tf	Void	Tie	Asigna un tiempo de fase al semáforo
Tf	Tie		Da el tiempo de fase del semáforo
f_nf	Void	Int	Asigna un número de fases al semáforo
Nf	Int		Da el número de fases del semáforo
agProh	Void	Prsem*	Agrega un bloqueo a lista de bloqueos del semáforo
Prh	Prsem*	Int	Da puntero al i-ésimo elemento de la lista de bloqueos
nPrh	Int		Da la cantidad de bloqueos de un semáforo
pase_sem	Bool	Via*, int	Determina si una vía no está bloqueada en una fase del semáforo

La estructura Prsem (prohibiciones del semáforo)

N	T	D
Viap	Via*	Guarda puntero a vía, la cual estaría bloqueada durante la fase que le acompaña
Fass	Ide	Guarda identificador de fase en el que está bloqueada la vía cuyo puntero le acompaña

Un bloqueo en un semáforo es un par (vía, fase) que indica que durante esa fase la vía está bloqueada por el semáforo.

La configuración de un semáforo contiene todos los bloqueos que realiza el semáforo durante todas sus fases.

El semáforo recuerda que vías están abiertas al tráfico guardando dentro de sus atributos una lista de pares con un puntero a una vía y un número de fase. Cada par informa que durante la fase dada la vía a la que apunta la primera componente

está cerrada al tráfico. Para guardar esta información se define una estructura de pares del tipo (puntero a vía, fase) de la cual la clase semáforo posee un vector para almacenar toda su configuración; se considera que desde el punto de vista de una vía un semáforo solamente tiene dos estados: verde, que permite el paso de los vehículos y rojo, que bloquea el paso de los mismos.

Los datos de todos los semáforos de la red se guardan por separado en dos archivos de texto. El primero llamado dsemaforos.txt que contiene por fila un identificador de semáforo, el número del nodo en que se ubica el semáforo, el número de fases del semáforo y el tiempo de fase del mismo. El segundo llamado psemaforos.txt que contiene por fila un número de fase y un identificador de vía que está bloqueada durante esa fase.

Clase nodo

La clase nodo tiene como atributos

N	T	D
id_	Ide	Identificador del nodo
xx_	Cor	Coordenada x del nodo
yy_	Cor	Coordenada y del nodo
sem_	Sem*	Puntero al semáforo del nodo
vvi	Vvi	Vector punteros a las vías de entrada al nodo
vvi	Vvi	Vector de punteros a las vías de salida del nodo

La clase nodo tiene como métodos

N	T	TA	D
f_id	Void	Ide	Asigna un identificador al nodo
Id	Ide	--	Da el identificador del nodo
f_xx	void	Cor	Asigna un valor a la coordenada x del nodo
Xx	Cor	--	Da el valor de la coordenada x del nodo
f_yy	void	Cor	Asigna un valor a la coordenada y del nodo
Yy	Cor	--	Da el valor de la coordenada y del nodo
a_sem	void	Sem*	Asigna un puntero al semáforo del nodo
Sem	Sem*	--	Da el puntero del semáforo del nodo
agViaent	void	Via*	Agrega puntero al vector de punteros a las vías de entrada del nodo
Vviaent	Via*	Int	Da el puntero a la i-ésima vía de entrada al nodo
nViaent	Int	--	Da el número de vías de entrada al nodo
agViasal	void	Via*	Agrega puntero al vector de punteros a las vías de salida del nodo
Vviasal	Via*	int	Da el puntero a la i-ésima vía de salida del

			nodo
nViasal	Int	--	Da el número de vías de salida del nodo
moverEnNodo	void	int&	Ejecuta el movimiento de vehículos a través del nodo

Cada nodo tiene asociado:

Un atributo que apunta al semáforo que se encuentra en el nodo, o en caso que no haya semáforo apunta a NULL.

Un vector de punteros que apuntan a las vías que llegan al nodo.

Un vector de punteros que apuntan a las vías que parten del nodo.

Clase vías

N	T	D
id_	Ide	Identificador de la vía
delNodo_	Nodo*	Puntero al nodo inicial de la vía
hacNodo_	Nodo*	Puntero al nodo final de la vía
lon_	Lon	Longitud de la vía
tasaent_[N P]	Tas	Tasa de entrada de los vehículos a la vía
vlibre_	bool	Establece si los vehículos en la vía están afectados por un alto o no
vgiros_	Vgi	Vector de giros que inician en la vía
cvia_[MV]	int	Cantidad de veces que han habido i vehículos en la vía
cpvvia_	double	Cantidad promedio de vehículos en la vía en cada momento
cvpvia_	int	Cantidad de vehículos que han transitado completamente la vía
tpvvia_	double	Tiempo promedio de recorrido de la vía de los vehículos que han pasado por la vía
tpavia_	double	Tiempo promedio que los vehículos que han circulado por la vía han esperado adelante
coorxccl_	VCo	Vector de coordenadas x de las celdas en que se divide la vía
cooryccl_	VCo	Vector de coordenadas y de las celdas en que se divide la vía
celdas_	Cel	Vector de punteros a los vehículos en las celdas de la vía

Métodos de la clase vía

N	T	TA	D
f_id	Void	Ide	Asigna un valor al identificador de la vía
Id	Ide	--	Da el identificador de la vía
f_delNodo	Void	Nodo*	Asigna un puntero al nodo inicial de la vía
delNodo	Nodo*	--	Da puntero al nodo inicial de la vía
f_hacNodo	Void	Nodo*	Asigna puntero al nodo final de la vía

hacNodo	Nodo*	--	Da puntero al nodo final de la vía
f_longitud	Void	Lon	Asigna longitud a la vía
Longitud	Lon	--	Da la longitud de la vía
f_tentrada	Void	Int, Tas	Asigna tasa de entrada a la vía en el período de tiempo i
Tentrada	Tas	Int	Da la tasa de entrada a la vía en el período de tiempo i
f_vlibre	Void	Bool	Asigna el valor booleano para saber si la vía tiene alto o no
vlibre_alt	Bool	--	Da el valor booleano que dice si la vía tiene alto o no
agGiro	Void	Giro*	Agrega puntero al vector de punteros a giros que inician en la vía
Vgiros	Giro*	Int	Da puntero al giro i que inicia en la vía
nVgiros	Int	--	Da el número de giros que inician en la vía
i_cvía	Void	--	Inicializa con ceros el vector que cuenta cantidad de veces que han habido i vehículos en la vía
f_cvía	Void	int, int	Asigna valor a la cantidad de veces que han habido i vehículos en la vía
Cvía	Int	Int	Da la cantidad de veces que han habido i vehículos en la vía
i_cpavía	Void	--	Inicializa en cero el promedio de vehículos en la vía
f_cpavía	Void	Double	Asigna valor al promedio de vehículos en la vía
Cpavía	Double	--	Da el promedio de vehículos en la vía
i_cpvía	Void	--	Inicializa en cero la cantidad de vehículos que han pasado la vía
f_cpvía	Void	Double	Asigna valor a la cantidad de vehículos que han pasado la vía
Cpvía	Int	--	Da la cantidad de vehículos que han pasado la vía
a_travía	Void	--	
i_tpavía	Void	--	Inicializa en cero el tiempo promedio de recorrido de los vehículos que han pasado por la vía
f_tpavía	Void	Double	Asigna valor al tiempo promedio de recorrido de los vehículos que han pasado por la vía
Tpavía	Double	--	Da el tiempo promedio de recorrido de los vehículos que han pasado por la vía
a_travía	Void	--	
i_tpavía	Void	--	Inicializa en cero el tiempo promedio que los vehículos que han circulado por la vía han esperado adelante
f_tpavía	Void	Double	Asigna valor al tiempo promedio que los vehículos que han circulado por la vía han esperado adelante
Tpavía	Double	--	Da el tiempo promedio que los vehículos que han circulado por la vía han esperado adelante

agv_celdas	Void	Veh*	Agrega puntero a vehículo en la última posición del vector de celdas de la vía
Celdas	Veh*	Int	Da el puntero al vehículo en la celda i de la vía
f_celda	Void	int, Veh*	Asigna puntero a vehículo a la celda i de la vía
nCeldas	Int	--	Da la cantidad celdas en que se divide la vía
meterAVía	Void	Veh*	Si la primera celda de la vía está vacía, asigna a ella puntero a vehículo
primeroEnVía	Veh*	--	Da puntero a vehículo en la celda delantera de la vía
quiPrimEnVía	Void	Veh*	Elimina puntero al vehículo que está en la celda final de la vía
tienEspacio	Bool	--	Determina si la primera celda de la vía apunta a un vehículo o no
moverEnVía	Void	int&	Ejecuta el movimiento de los vehículos a lo largo de la vía sin incluir la celda final de la vía
ag_CoorXCel	Void	Cor	Agrega una coordenada x al vector de coordenadas x de las celdas de la vía
ag_CoorYCel	Void	Cor	Agrega una coordenada y al vector de coordenadas x de las celdas de la vía
CoorXCel	Cor	Int	Da la coordenada x de la celda i de la vía
CoorYCel	Cor	Int	Da la coordenada x de la celda i de la vía
inicializar_vía	Void	--	Inicializa el estado de la vía
estado_vía	Void	--	Imprime información sobre la situación de los vehículos en la vía
Rendvia	Void	--	Calcula medidas de rendimiento de la vía

Clase Giro

Un giro es un par de punteros, el primero a la vía desde donde puede circular un vehículo y el segundo a la vía en que podría continuar circulando el vehículo. Un giro caracteriza el comportamiento de los vehículos al pasar una intersección.

Atributos de la clase Giro

N	T	D
viaen_	Vía*	Puntero a la vía de entrada al giro
viasa_	Vía*	Puntero a la vía de salida del giro
frecu_[NP]	Pro	Frecuencias de campo del giro en cada período de la simulación
ngirosim_[NP]	int	Número calculado de veces del giro en la simulación en cada período
frecusim_[NP]	Pro	Frecuencia calculada del giro en la simulación en cada período

Métodos de la clase Giro

N	T	TA	D
f_viaen	void	Via*	Asigna puntero a la vía de entrada del giro
viaengiro	Via*	--	Da puntero a la vía de entrada del giro
f_viasa	void	Via*	Asigna puntero a la vía de salida del giro
Viasagiro	Via*	--	Da el puntero a la vía de salida del giro
f_frecu	void	int, Pro	Asigna valor a la frecuencia de campo del giro durante el período i de la simulación
Frecugiro	Pro	int	Da la frecuencia de campo del giro durante el período i de la simulación
i_ngirosim	void	int	Inicializa en cero el número calculado de veces que se ha realizado el giro en la simulación
a_ngirosim	void	int	Incrementa en uno la cantidad calculada de veces que se ha realizado el giro en la simulación
f_frecusim	void	int, Pro	Asigna valor a la frecuencia de campo del giro durante el período i de la simulación
Frecusim	Pro	int	Da la frecuencia calculada del giro durante el período i de la simulación
Ngirosim	int	int	Da el número calculado de veces que se ha realizado el giro en la simulación durante el período i

Clase simulación

N	T	TA	D
Simcirculacion	Void	bool&	
cal_frecesim	Void	int	
leerNodos	Void	--	
leerVias	Void	--	
leerGiros	Void	--	Función para leer los giros en la red y sus frecuencias
leerDaSem	Void	--	Función para leer los semáforos de la red
leerPrSem	Void	--	Función para leer los semáforos de la red

5.4.6. ENTRADA DE LA SIMULACIÓN

Como datos de entrada para la simulación se requiere información acerca de los nodos, las vías, los semáforos y la frecuencia de los giros en cada nodo, para ello se tienen los siguientes archivos:

- Un archivo llamado nodos, este archivo contiene, un identificador de nodos con las coordenadas x y y, también un identificador para determinar si es nodo de entrada o nodo de salida de la red.

Formato de archivo de nodos

ID	ESTE	NORTE	ENT	SAL
1	513479.68	1375307.76	1	0
2	513483.53	1375209.02	1	1
⋮	⋮	⋮	⋮	⋮
59	512856.26	1374417.63	1	0

- Un archivo llamado vía, este archivo contiene, un identificador de vía, con un nodo inicial y un nodo final, además contiene la información acerca de si la vía posee algún tipo de señal de tránsito.

Formato de archivo de vías

ID	NODOINICIO	NODOFINAL	PRIORIDAD
1	1	2	1
2	2	3	1
⋮	⋮	⋮	⋮
103	59	58	0

- Un archivo llamado dsemaforo, este archivo contiene los datos de los semáforos en la red, estos son: un identificador del semáforo, el nodo donde está ubicado el semáforo, el tiempo de fase y el número de fases que tiene el semáforo.

Formato de archivo de semáforo

ID	POS	TF	NF
1	35	5	2

- Un archivo llamado psemaforo, este archivo contiene la información acerca de la vía que tiene prioridad de pase en cada fase.

Formato de archivo de datos de prioridad de semáforos

ID	FASE	EFASE	SFASE
1	0	52	59
1	0	52	68
1	1	60	59
1	1	60	68

- Un archivo llamado giros, este archivo contiene las vías inicial y final de todos los giros de la red y las frecuencias con que se dan estos giros en cada período del día.

5.4.7. SALIDA DE LA SIMULACIÓN (RESULTADOS)

La simulación se da mediante un programa C++, el cual contiene otros elementos distintos de los aquí mencionados, por ejemplo funciones para lectura de datos y obtención de resultados

Tiempo	Id vehicular	Coordenada x	Coordenada y
1	1	730.0013901	999.5162625
1	2	733.5343703	900.7700012
.	.	.	.
.	.	.	.
.	.	.	.
1	90	105.8353829	125.8690655
2	1	730.2156502	994.0204375
2	2	733.5372839	895.2700019
.	.	.	.
.	.	.	.
.	.	.	.
2	90	105.5523048	131.3617758
.	.	.	.
.	.	.	.

La tabla anterior es un archivo de salida de la simulación (arimagen), que nos da información acerca de la posición de cada vehículo en la red en cada momento de la simulación. Al paso de una unidad de tiempo los vehículos se mueven cambiando así sus posiciones, siendo estas calculadas y actualizadas por el programa C++ de la simulación, de esta manera podemos darle seguimiento a un vehículo para observar su recorrido y las posibles dificultades que podrían presentársele. Por ejemplo podemos observar que el vehiculo1 en el momento 1 tiene coordenadas $x=730.0013901$, $y=999.5162625$ después de una unidad de tiempo sus nuevas coordenadas son: $x=730.2156502$, $Y=994.0204375$.

De igual forma da como salidas archivos que contienen la cantidad de vehículos en la vía, el tiempo que tarda un vehículo en recorrer la vía y la probabilidad de encontrar n vehículos en la vía.

5.4.8. VISUALIZACIÓN DE LA SIMULACIÓN.

La visualización de los resultados se realiza mediante la herramienta de programación matemática MATLAB, el cual mediante sus funciones nos permite escribir un programa capaz de graficar la red de tráfico a escala real y además representar los vehículos en la red de forma dinámica.

Para lograr la visualización como datos de entrada se toman los datos de salida del programa creado en C++, estos datos de salida son los correspondientes a las

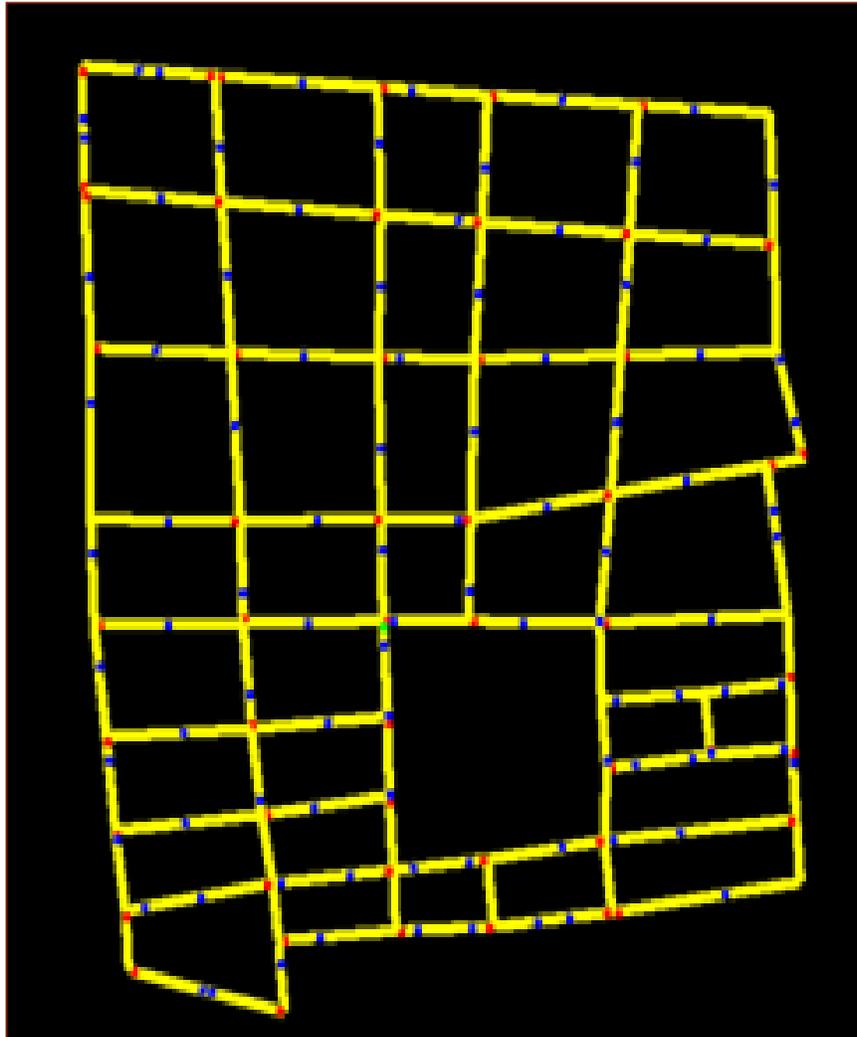
coordenadas x y y de los vehículos en cada iteración, así como la ubicación de semáforos y los tiempos de fase de éstos.

El programa creado en MATLAB pone en cada momento las coordenadas actualizadas de los vehículos de este modo se observa el movimiento de los vehículos siguiendo las leyes de circulación establecidas al momento de crear el programa en C++.

De igual manera sitúa los semáforos en las coordenadas establecidas y también representa los tiempos de fase con verde si es pase o rojo si es alto.

Todo esto con el objeto de representar el tráfico de la forma más real posible y de ahí hacer comparaciones con el comportamiento real de los vehículos en la red.

La siguiente figura corresponde a la visualización de una iteración del programa; los puntos azules representan vehículos que se mueven por unidad de tiempo, los puntos rojos representan las señales de alto asociado al nodo correspondiente



5.5. LA VALIDACIÓN DE LA SIMULACIÓN

Una etapa importante de nuestro trabajo consiste en el estudio del comportamiento del programa respecto al desempeño real de la red estudiada. Para ello se eligieron cuatro vías de manera aleatoria para calcular sus respectivas medidas de desempeño; la primera parte son medidas generadas por el programa y en una segunda parte y en contraste con la primera, medidas de desempeño tomadas en tiempo real.

5.5.1. MEDIDAS DE DESEMPEÑO EN BASE A LA SIMULACIÓN.

El programa fue creado con la cualidad de presentar las medidas de desempeño hasta cualquier momento de la simulación, gracias a códigos definidos en la programación. Dichas medidas nos muestran el comportamiento de la red de tráfico y además nos otorgan datos explícitos para hacer nuestro respectivo análisis contrastándolos con datos de la red real estudiada.

Como principales medidas de rendimiento tenemos:

- Tiempo promedio del vehículo en recorrer la vía
- Número promedio de vehículos en la vía
- Probabilidad de encontrar n vehículos en la vía

A continuación mostramos parte de los archivos generados por el programa para cada una de las medidas de rendimiento anteriores:

Tiempo promedio del vehículo en recorrer la vía (guardados en el archivo de salida "trecvia")

Momento	Vía	Tiempo Promedio	No. vehículos
22	34	12	1
26	1	16	1
31	2	5	1
32	34	12	2
36	1	16	2
.	.	.	.
.	.	.	.
.	.	.	.

La tabla anterior se lee de la siguiente manera:

En el momento 22, en la vía 34 pasa el primer vehículo con tiempo promedio de 12 segundos y así sucesivamente.

Número promedio de vehículos en la vía (guardados en el archivo de salida "cvehia")

Vía	Momento	Cantidad promedio de vehículos en la vía
1	1	0.1818
1	2	0.25
1	3	0.3077
.	.	.
.	.	.
.	.	.
1	15000	0.4706
2	1	0.5
2	2	0.5263
.	.	.
.	.	.
.	.	.
2	1500	0.4375
.	.	.
.	.	.
.	.	.

103	1	0.6
103	2	0.3571
103	3	0.4
.	.	.
.	.	.
.	.	.
103	15000	0.8333

Probabilidad de encontrar n vehículos en la vía (almacenados en el archivo de salida "probest").

Vía	Momento	Po	P1	P2	. . . Pn
1	1	0.2354	0.3658	0.2394	. . .
1	2	0.2353	0.3658	0.2394	. . .
1	3	0.2353	0.3659	0.2393	. . .
.
.
.
1	15000	0.2354	0.4721	0.2393	. . .
2	1	0.000744	0.6672	0.3065	. . .
2	2	0.000744	0.6672	0.3065	. . .
2	3	0.0007439	0.6671	0.3066	. . .
.
.
.
2	15000	0.0007438	0.6671	0.3066	. . .
.
.
.
103	1	0.2173	0.34	0.2503	. . .
103	2	0.2173	0.34	0.2503	. . .
.
.
.
103	15000	0.2245	0.39	0.2741	. . .

Donde cada momento se entiende como unidad de tiempo, teniendo como límite de la simulación para este caso 15000 momentos, y P0 es la probabilidad de encontrar 0 vehículos en la vía; P1 es la probabilidad de encontrar 1 vehículo en la vía y así sucesivamente.

5.5.2. MEDIDAS DE DESEMPEÑO EN BASE A DATOS REALES

Para hacer comparaciones entre la simulación y la red real se procedió a tabular datos obtenidos mediante trabajo de campo sobre las medidas de rendimiento real

requeridas en las vías 16, 35, 65, 75. En este caso la herramienta fundamental fueron grabaciones en videos del flujo vehicular en estas vías en períodos de tiempo establecidos, siendo estos de 20 minutos por la mañana con el objeto de tomar y estudiar de manera más exacta el comportamiento de los vehículos.

Debido a que la información procesada en estas vías es demasiado amplia, solamente presentamos como ejemplo los estudios realizados sobre la vía 16.

5.5.3. CÁLCULO DE MEDIDAS DE DESEMPEÑO REALES PARA LA VÍA 16.

Luego de levantar los videos procedimos a tomar el tiempo de entrada a la vía, el tiempo de llegada al final de la vía y el tiempo de salida de la vía (cuando el vehículo cruza la intersección) para cada vehículo y en base a esta información ya organizada y completa se procedió a realizar los cálculos respectivos, obteniéndose así los siguientes resultados:

Tiempo promedio del vehículo en recorrer la vía

No. de vehículos	Tiempo en la vía	Momento	Promedio
1	21	44	21
2	30	59	26
3	32	64	28
.	.	.	.
.	.	.	.
.	.	.	.
94	38	1134	39

Estos indicadores fueron tomados de la siguiente manera:

No. de vehículos (N): cantidad de autos que ingresaron a la vía durante este período de tiempo.

Tiempo en la vía (T): tiempo en segundos que tarda el vehículo en recorrer la vía desde su entrada hasta su salida.

Momento (M): es considerado como el instante en el que cada vehículo llega al final de la vía.

Promedio (P): es el promedio simple recurrente de, los tiempos en la vía, donde:

$$P_n = \frac{P_{n-1} + P_{n-2} + \dots + P_0}{n} ; P_0 = T_0$$

Número promedio de vehículos en la vía

Momento	No. de vehículos	Promedio
0	0	0.0000
1	0	0.0000
2	0	0.0000
.	.	.
.	.	.
23	1	0,0435
24	1	0,0833
.	.	.
.	.	.
30	2	0.3333
.	.	.
.	.	.
32	3	0.4688
.	.	.
.	.	.
40	4	1.0000
.	.	.
.	.	.
572	3	3.1016
.	.	.
.	.	.
580	5	3.1155
.	.	.
.	.	.
603	2	3.1443
.	.	.
.	.	.
.	.	.
1139	1	3.6130

En este caso definimos los indicadores como:

Momento (M): cada una de las unidades de tiempo transcurridos desde que inicia hasta que finaliza el período.

No. de vehículos (N): cantidad de vehículos presentes en la vía en cada unidad de tiempo.

Promedio (P): cantidad promedio de vehículos en el sistema en cada unidad de tiempo, calculado de la siguiente manera:

$$P_n = \frac{M_{n-1} * P_{n-1} + N_n}{M_n}; \text{ con } P_0 = N_0 = M_0 = 1$$

Probabilidad de encontrar n vehículos en la vía

MOMENTO	P0	P1	P2	P3	P4	P5	P6
23	0,0000	0,2609	0,0000	0,0000	0,0000	0,0000	0,0000
29	0,0000	0,2069	0,1034	0,0000	0,0000	0,0000	0,0000
32	0,0000	0,1875	0,0938	0,2500	0,0000	0,0000	0,0000
40	0,0000	0,1500	0,0750	0,2000	0,1250	0,0000	0,0000
45	0,0000	0,1333	0,0667	0,1778	0,1111	0,2000	0,0000
54	0,0000	0,1111	0,0556	0,1481	0,1481	0,1667	0,0000
57	0,0000	0,1053	0,0526	0,1404	0,1404	0,2281	0,0000
61	0,0000	0,0984	0,0492	0,1311	0,1311	0,2131	0,0328
.
.
.
1139	0,0105	0,0500	0,1370	0,2239	0,2994	0,1633	0,0711

El cálculo de estas medidas se realizó de forma diferente, inicialmente se encontró la longitud de cada período de tiempo en los cuales el número de vehículos se mantuvo constante por medio de una diferencia simple entre los momentos de entrada de dos vehículos consecutivos.

Simultáneamente se calculó la cantidad de vehículos en la vía dentro de cada período.

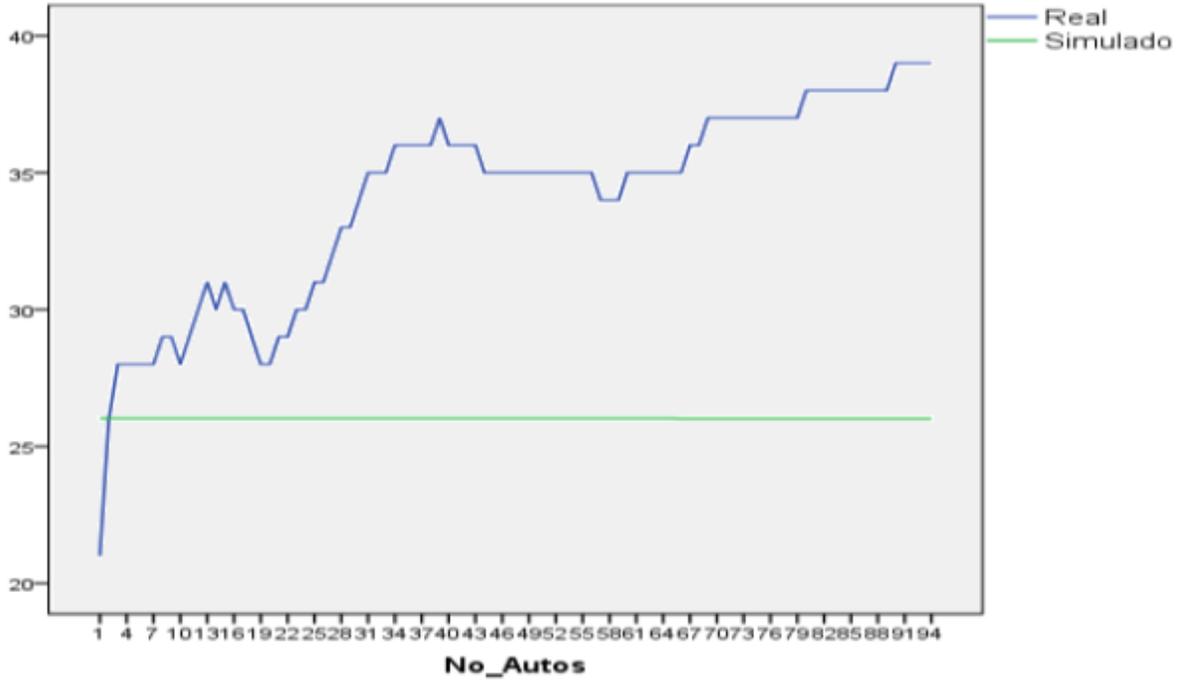
Finalmente para calcular las probabilidades de encontrar 1, 2, 3,..., n vehículos en la vía se contó en cada momento las veces que aparecían el mismo número de vehículos y se dividió la suma de las respectivas longitudes de período por el momento dado, dando como resultado la tabla anterior.

5.5.4. COMPARACIÓN DE MEDIDAS DE DESEMPEÑO

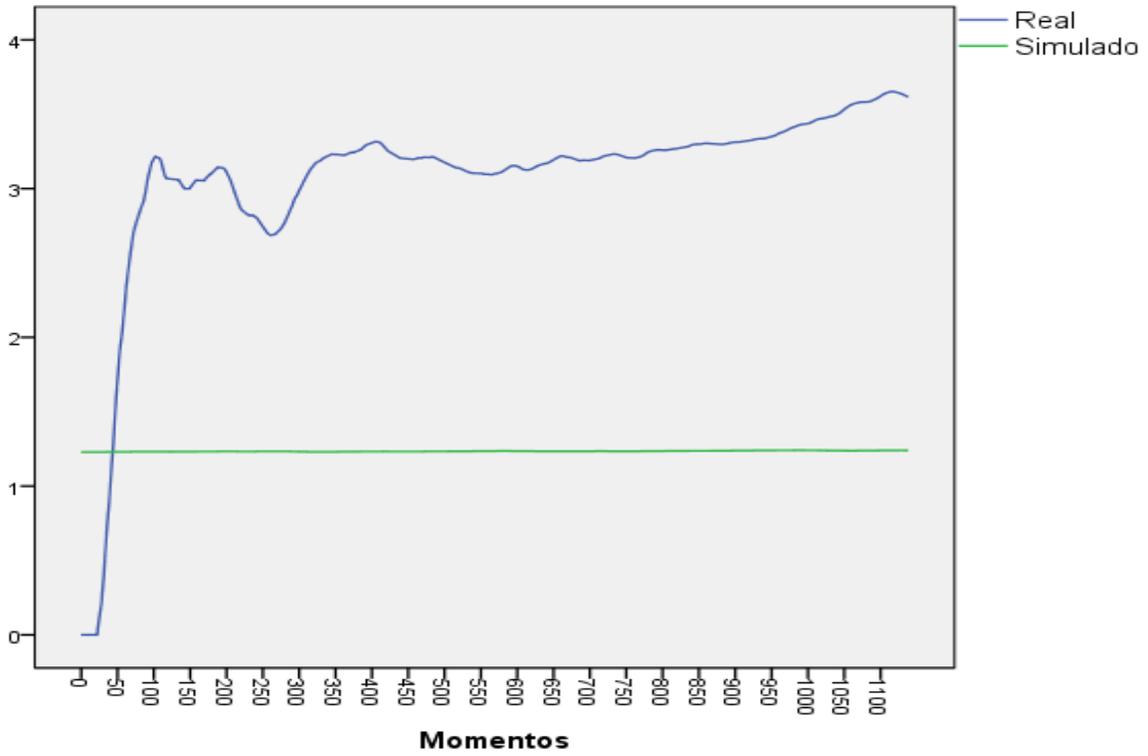
Para poder realizar una comparación adecuada entre datos reales y simulados procedimos a seleccionar de las medidas de rendimiento generadas por la simulación toda la información concerniente a las vías a validar. Luego organizamos esta información en tablas similares a las tablas de las medidas de desempeño de los datos reales; teniendo el cuidado de seleccionar los tiempos y los momentos correspondientes a los datos reales, finalmente procedimos a realizar comparaciones entre ambas medidas por medio de gráficos de secuencia.

A continuación presentamos los gráficos resultantes de las comparaciones realizadas a las medidas de desempeño de la vía 16.

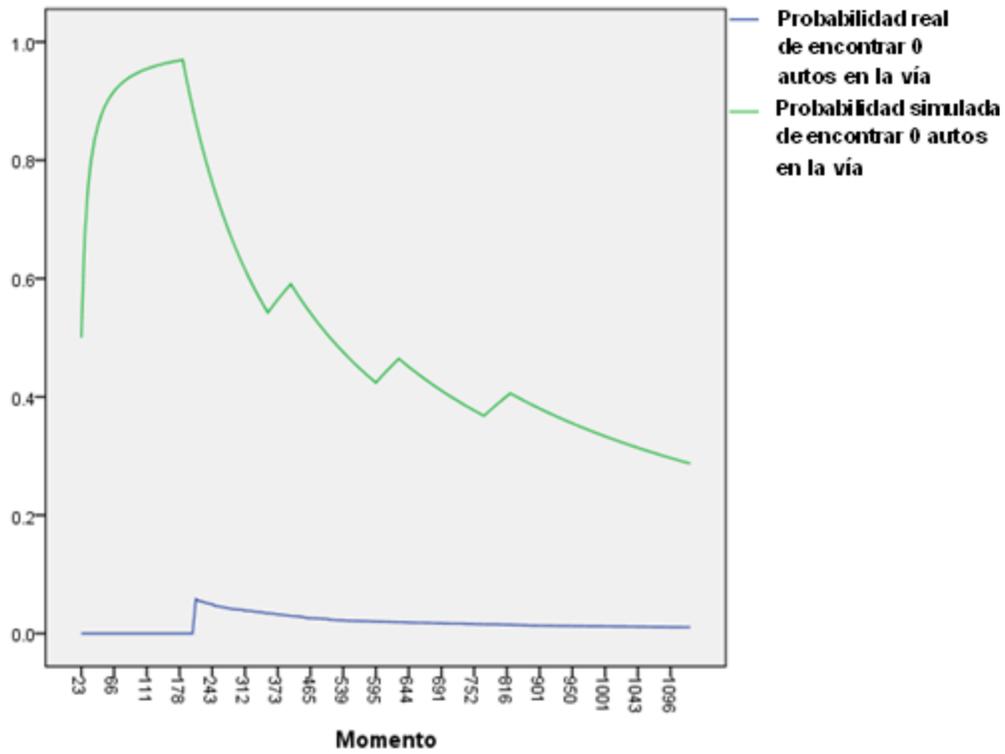
TIEMPO PROMEDIO EN LA COLA



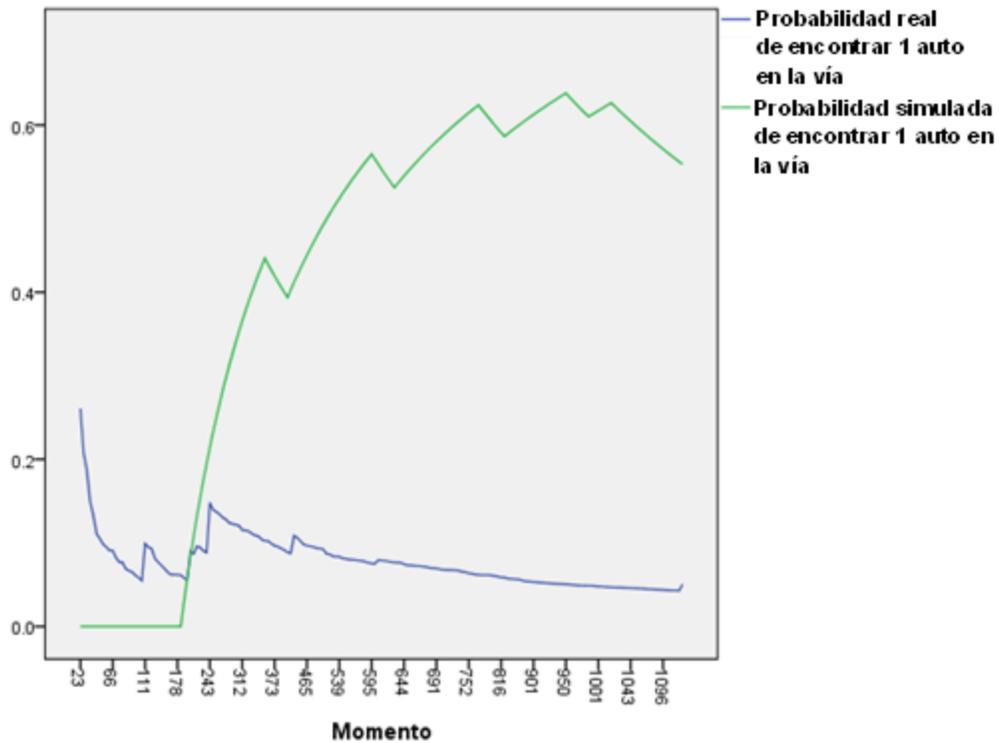
NÚMERO PROMEDIO DE CLIENTES EN EL SISTEMA



PROBABILIDAD DE ENCONTRAR 0 AUTOS EN LA VÍA



PROBABILIDAD DE ENCONTRAR 1 AUTO EN LA VÍA





Es evidente observar una diferencia considerable en el comportamiento de las medidas de rendimiento calculadas en la simulación y las medidas de rendimiento calculadas de forma real.

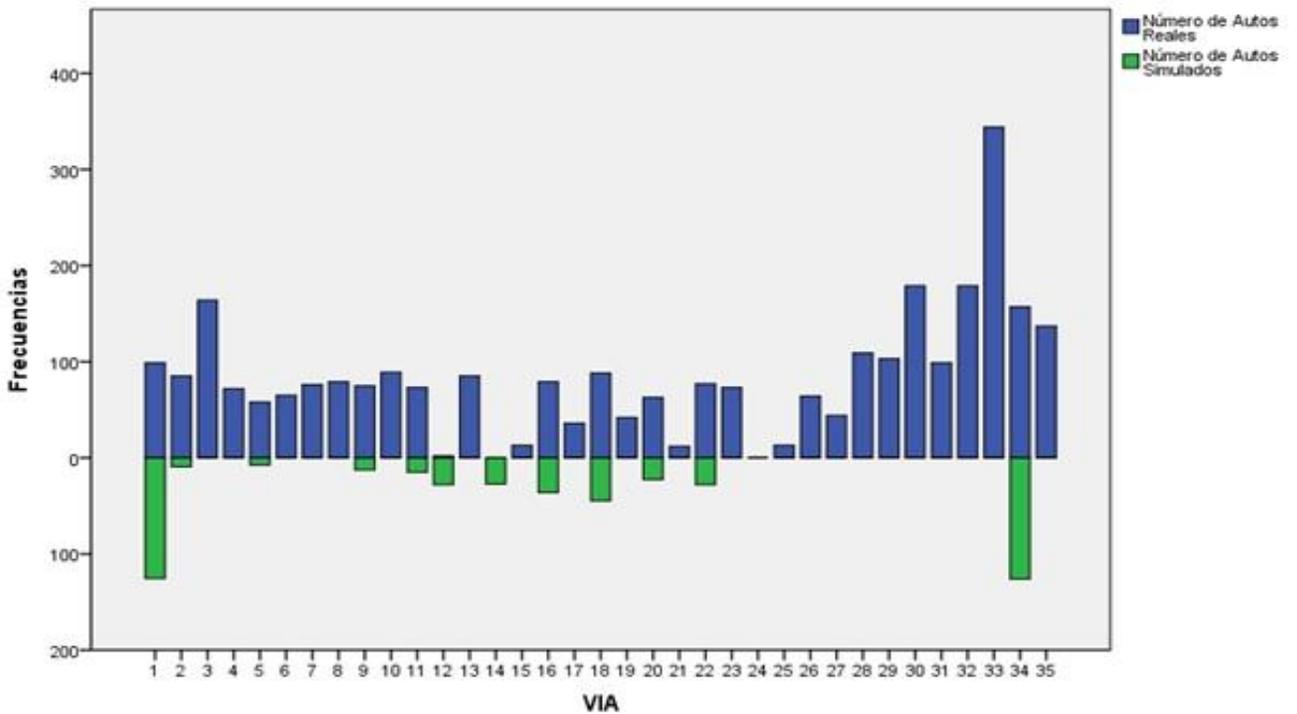
Este comportamiento se mantuvo en los análisis correspondientes a las medidas de rendimiento de las vías 35, 65 y 75.

En busca a una justificación a este comportamiento se procedió a realizar comparaciones en dos parámetros que no habían sido tomados en consideración como son: el número de autos de entrada a la vía y la tasa de entrada de autos a la vía por minutos.

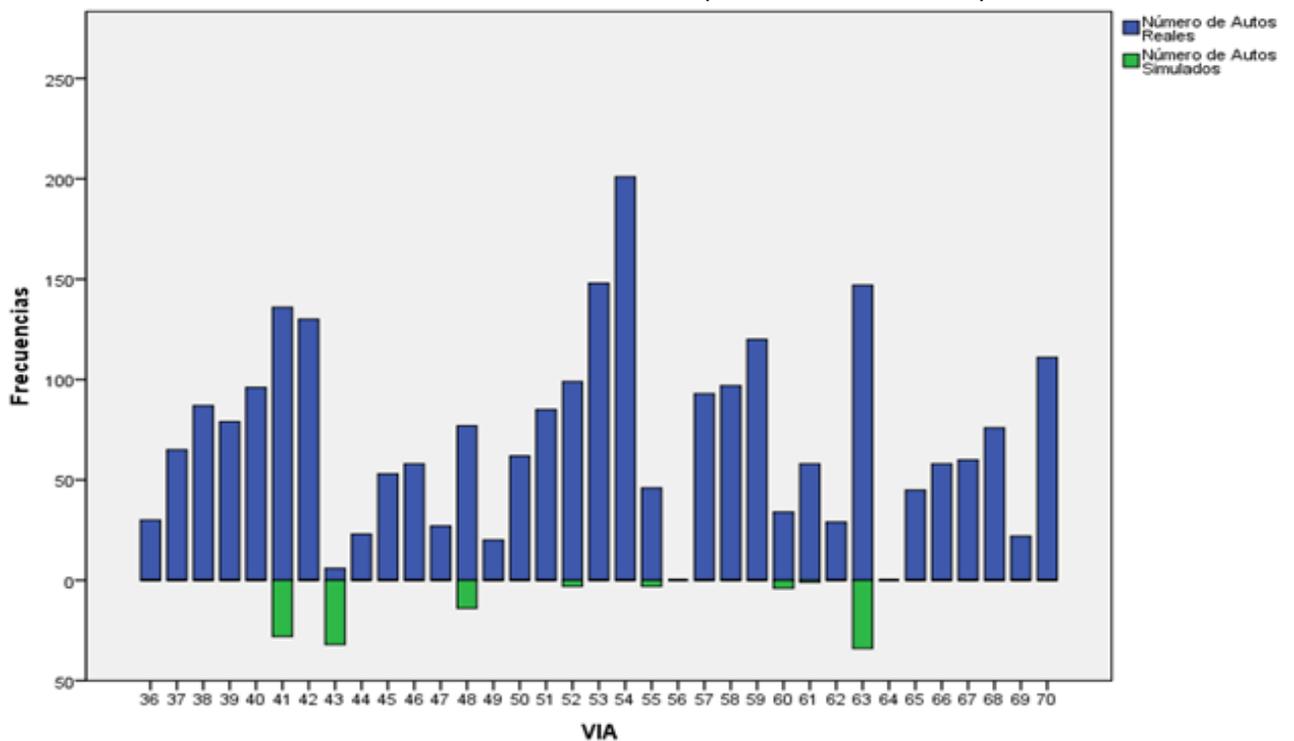
El estudio de estos dos parámetros se realizó sobre cada una de las vías por lo que fue necesario realizar subdivisiones de estas, generando de este modo tres gráficos para cada parámetro estudiado.

Dichas comparaciones tuvieron como resultado los siguientes gráficos:

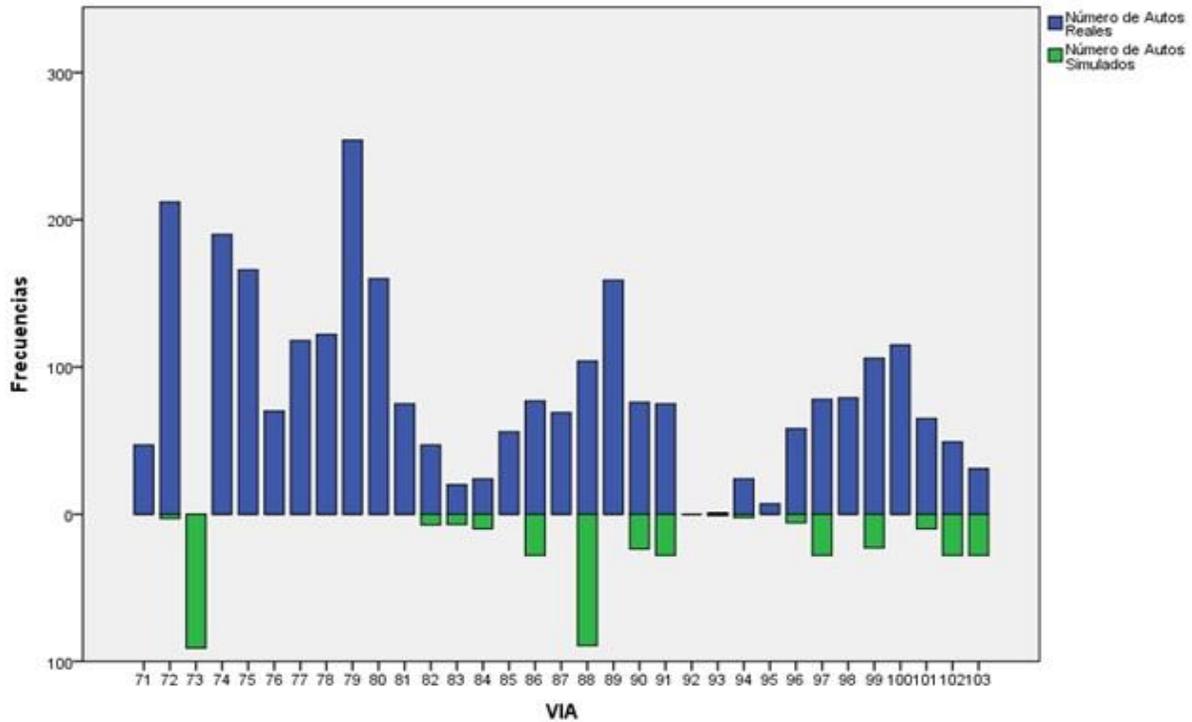
NÚMERO REAL DE AUTOS DE ENTRADA A LA VÍA VS NÚMERO SIMULADOS DE AUTOS DE ENTRADA A LA VÍA (PRIMERA PARTE)



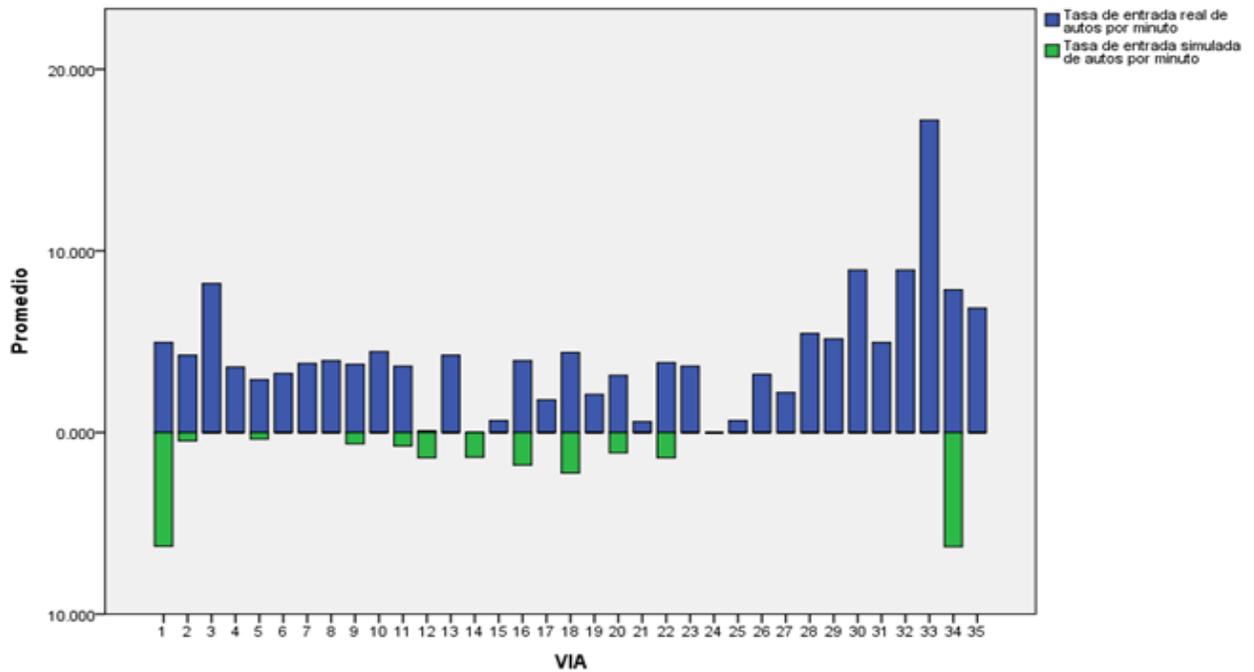
NÚMERO REAL DE AUTOS DE ENTRADA A LA VÍA VS NÚMERO SIMULADOS DE AUTOS DE ENTRADA A LA VÍA (SEGUNDA PARTE)



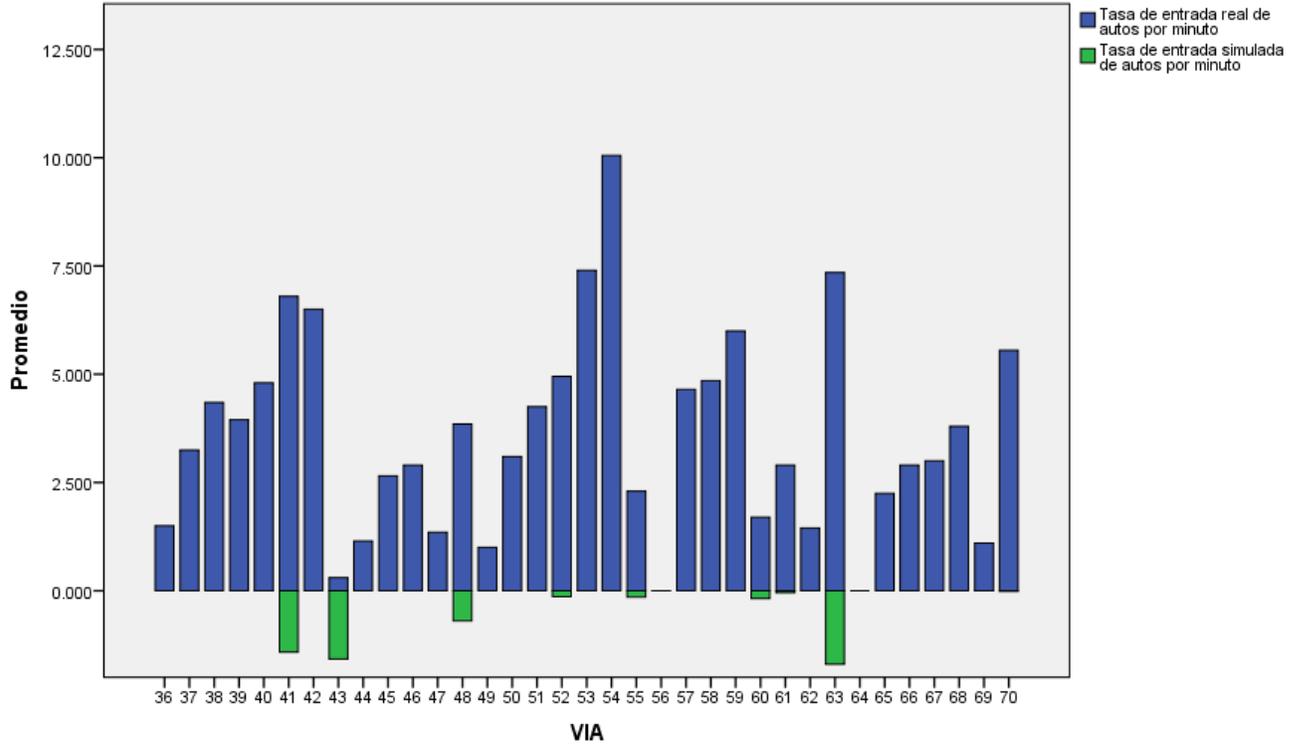
NÚMERO REAL DE AUTOS DE ENTRADA A LA VÍA VS NÚMERO SIMULADOS DE AUTOS DE ENTRADA A LA VÍA (TERCERA PARTE)



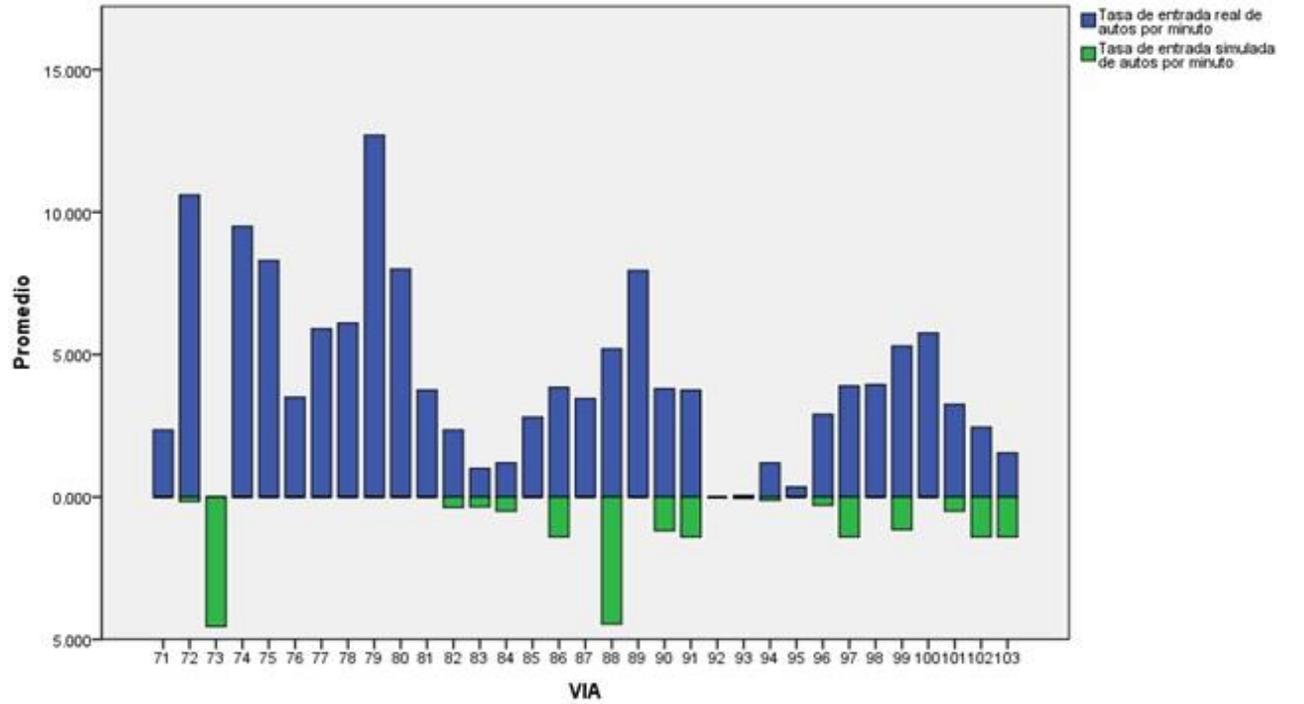
TASA DE ENTRADA REAL DE AUTOS POR MINUTO VS TASA DE ENTRADA SIMULADA DE AUTOS POR MINUTO (PRIMERA PARTE)



TASA DE ENTRADA REAL DE AUTOS POR MINUTO VS TASA DE ENTRADA SIMULADA DE AUTOS POR MINUTO (SEGUNDA PARTE)



TASA DE ENTRADA REAL DE AUTOS POR MINUTO VS TASA DE ENTRADA SIMULADA DE AUTOS POR MINUTO (TERCERA PARTE)



Los gráficos anteriores, tanto número de entrada de autos a la vía como la tasa de entrada de autos a la vía muestran que la cantidad de autos que entran en una calle por unidad de tiempo en la realidad no es igual a la cantidad de autos que entran en la vía durante la simulación, lo cual consideramos puede ser una causa de las diferencias presentadas en los gráficos de comparación de las medidas de rendimiento.

6. CONCLUSIONES

Con nuestro trabajo tratamos de realizar una caracterización completa de todos los actores involucrados en el sistema de tráfico de la ciudad de León (vías, nodos semáforos, intersecciones, vehículos.) y sus interacciones, de tal manera que podemos describir el funcionamiento de dicho sistema.

En base a los resultados obtenidos de la simulación y la comparación con datos reales destacamos lo siguiente:

- ❖ La validación de la presente simulación de tráfico se considera pendiente, debido a la magnitud de las diferencias observadas al momento de comparar las medidas de rendimiento reales con las simuladas. y por ende no hay garantía de que la experimentación con esta simulación refleje en buena medida el comportamiento del sistema de tráfico real.
- ❖ En la simulación el tiempo promedio que tardan los vehículos en cruzar la vía es menor que los obtenidos en los datos reales, esto, por un desajuste en las velocidades de los vehículos en la simulación y a interrupciones de la fluidez de los vehículos en el sistema real (parada de algunos taxis o rutas a bajar pasajeros).
- ❖ En los datos reales el tiempo promedio que tarda un vehículo en cruzar la intersección (el cliente es atendido) es mucho mayor que los arrojados por la simulación, esto puede ser ocasionado por las interrupciones de algunos vehículos con prioridades especiales como el camión recolector de basura o vehículos pasajeros.
- ❖ Las probabilidades de encontrar n vehículos en la vía en el sistema real y en la simulación son muy diferentes, producto de la propagación de las grandes diferencias observadas en las cantidades reales y simuladas de vehículos presentes en cada vía.
- ❖ En la simulación la velocidad es tomada como constante y los autos tienen restricciones como la de no adelantar, no estacionarse y tener un tamaño promedio establecido lo que también es causa de las grandes variantes al momento de contrastar las medidas de rendimientos reales con las simuladas.

7. RECOMENDACIONES

Realizando las consideraciones necesarias y sujetos a los resultados obtenidos recomendamos lo siguiente:

- ❖ Considerar versiones más sofisticadas de simulación de tráfico, que incluyan factores que hasta la fecha no han sido considerados, de tal forma que puedan disminuirse las discrepancias existentes entre el sistema real y el sistema simulado.
- ❖ Realizar trabajos de campos con periodos de tiempos más prolongados y fechas sincronizadas para obtener datos más apegados a la realidad y resultados más efectivos.
- ❖ Tomar las medidas de rendimiento a un número mayor de vías con un número mayor de ensayos, esto, para realizar una comparación y evaluación de las mismas de manera que se realice un proceso de análisis más efectivo en la red de tráfico de estudio.
- ❖ Realizar una estimación de la cantidad promedio de vehículos que entran y se parquean dentro de la red por cada una de las vías y considerar esta en la simulación.
- ❖ Dotarse de dispositivos electrónicos de tal manera que se puedan obtener datos exactos para el tráfico vehicular que puedan incluirse en la simulación y realizar los respectivos análisis.
- ❖ Trabajar de la mano con instituciones encargadas de la regulación del tráfico (alcaldías, policía, ministerio del transporte, cooperativas de transporte, etc.) como estrategia que facilite cualquier mejora al modelo de simulación de tráfico actual.
- ❖ Gestionar recursos económicos con instituciones interesadas o beneficiadas con un trabajo de este tipo, con el objeto de crear un fondo para solventar gastos presentes durante todo el proceso del trabajo de campo como son: la adquisición de equipos especializados y de personal que colabore en el proceso de recolección de la información necesaria para la validación.

8. BIBLIOGRAFÍA

- ❖ Investigación de Operaciones, 7a. edición, Hamdy A. Taha
- ❖ Multi-agent transportation simulation, Kail Nagel, February 2, 2004.
- ❖ Simulación estadística de eventos discretos en tráfico vehicular, Monografía, Teresa Martínez Flores.
- ❖ Theory and problems of Programming with c++, Second Edition, John R. Hubbard, Ph. D.
- ❖ Programación orientada a objetos con C++. 2a. edición 1997. Ceballos, F.J.
- ❖ Análisis del flujo vehicular.
<http://sjnavarro.files.wordpress.com/2008/08/analisis-de-flujo-vehicular-cal-y-mayor.pdf>.
- ❖ Elementos existentes en la teoría de cola
<http://file:///F:/Monograf%C3%ADa%20Amilcar/notacion%20kendall.htm>
- ❖ Notación de Kendal.
<http://www.auladeeconomia.com/L%EDneas%20de%20Espera.ppt>.
- ❖ Autómata Celular.
<http://2006.igem.org/wiki/images/1/1b/IntoCA.pdf>

9.2. FRECUENCIAS DE GIRO EN LOS NODOS DE LA RED

NODO	VINI	VFIN	P1	P2	P3	P4	P5	P6	...	P18
1	0	1	0.88393	0.81739	0.88618	0.80380	0.82317	0.81333	...	0.80916
1	0	15	0.11607	0.18261	0.11382	0.19620	0.17683	0.18667	...	0.19084
2	0	2	0.03922	0.08911	0.05042	0.08571	0.06711	0.05797	...	0.05102
2	1	0	0.16667	0.19802	0.25210	0.16429	0.16107	0.19565	...	0.15306
2	1	2	0.79412	0.71287	0.69748	0.75000	0.77181	0.74638	...	0.79592
3	2	3	0.51829	0.53289	0.59732	0.50649	0.55310	0.51628	...	0.46893
3	16	3	0.48171	0.46711	0.40268	0.49351	0.44690	0.48372	...	0.53107
4	3	0	0.56098	0.50980	0.48980	0.51754	0.52607	0.65019	...	0.45455
4	3	4	0.43902	0.49020	0.51020	0.48246	0.47393	0.34981	...	0.54545
5	0	5	0.09574	0.06542	0.09009	0.08442	0.06422	0.04762	...	0.05970
5	0	17	0.29787	0.24299	0.33333	0.27922	0.27523	0.30159	...	0.33582
5	4	5	0.52128	0.57944	0.44144	0.46753	0.49541	0.51587	...	0.48507
5	4	17	0.08511	0.11215	0.13514	0.16883	0.16514	0.13492	...	0.11940
6	5	0	0.05556	0.06923	0.06667	0.05233	0.06790	0.06587	...	0.05229
6	5	6	0.45139	0.37692	0.45185	0.49419	0.48765	0.52096	...	0.42484
6	7	0	0.49306	0.55385	0.48148	0.45349	0.44444	0.41317	...	0.52288
7	6	8	0.38710	0.31298	0.40678	0.47895	0.45026	0.36957	...	0.41509
7	18	7	0.49032	0.58015	0.48023	0.34211	0.31937	0.40761	...	0.40252
7	18	8	0.12258	0.10687	0.11299	0.17895	0.23037	0.22283	...	0.18239
8	0	9	0.11111	0.10909	0.11881	0.08571	0.05392	0.05473	...	0.12676
8	0	19	0.29060	0.26364	0.32673	0.22286	0.23529	0.18408	...	0.25352
8	8	9	0.52991	0.54545	0.41584	0.49714	0.41176	0.40299	...	0.38028
.
.
.
59	92	103	0.00000	0.00000	0.10811	0.10169	0.15385	0.13889	...	0.14286

9.3. TIEMPOS DE ENTRADA, LLEGADA Y SALIDA DE LOS AUTOS EN LA VÍA 16

NO. AUTO	ENTRADA	LLEGADA	SALIDA
1	0:00:23	0:00:44	0:00:54
2	0:00:29	0:00:59	0:01:03
3	0:00:32	0:01:04	0:01:06
4	0:00:40	0:01:08	0:01:13
5	0:00:45	0:01:16	0:01:18
6	0:00:57	0:01:21	0:01:32
7	0:01:01	0:01:33	0:01:38
8	0:01:06	0:01:39	0:01:43
9	0:01:18	0:01:44	0:01:49
10	0:01:27	0:01:47	0:01:51
11	0:01:30	0:02:11	0:02:15
12	0:01:56	0:02:34	0:02:39
13	0:01:58	0:02:42	0:02:57
14	0:02:23	0:02:46	0:02:58
15	0:02:30	0:03:06	0:03:09
16	0:02:50	0:03:12	0:03:17
17	0:02:57	0:03:18	0:03:20
18	0:02:59	0:03:23	0:03:27
19	0:03:39	0:03:52	0:03:57
20	0:03:42	0:03:58	0:04:03
21	0:03:52	0:04:49	0:04:53
22	0:04:18	0:04:53	0:04:56
23	0:04:22	0:04:57	0:05:12
24	0:04:28	0:05:12	0:05:16
25	0:04:36	0:05:17	0:05:24
26	0:04:41	0:05:29	0:05:33
27	0:04:51	0:05:43	0:05:46
28	0:04:58	0:05:47	0:05:52
29	0:05:13	0:06:07	0:06:13
30	0:05:31	0:06:24	0:06:33
31	0:05:51	0:06:39	0:06:46
32	0:06:03	0:06:42	0:06:51
33	0:06:16	0:06:52	0:06:54
34	0:06:26	0:07:33	0:07:45
35	0:07:03	0:07:50	0:07:53
36	0:07:20	0:07:59	0:08:03

37	0:07:39	0:08:04	0:08:06
38	0:07:49	0:08:37	0:08:41
39	0:08:00	0:08:53	0:08:58
40	0:08:37	0:09:10	0:09:13
41	0:08:56	0:09:18	0:09:22
42	0:08:59	0:09:24	0:09:31
43	0:09:15	0:09:39	0:09:52
44	0:09:23	0:09:53	0:09:55
45	0:09:26	0:09:57	0:09:59
46	0:09:33	0:10:01	0:10:04
47	0:09:40	0:10:24	0:10:27
48	0:10:07	0:10:28	0:10:30
49	0:10:11	0:10:34	0:10:55
50	0:10:16	0:10:56	0:10:59
51	0:10:22	0:10:59	0:11:02
52	0:10:28	0:11:03	0:11:05
53	0:10:44	0:11:10	0:11:14
54	0:10:55	0:11:27	0:11:31
55	0:11:09	0:11:44	0:11:47
56	0:11:26	0:11:48	0:11:51
57	0:11:28	0:11:57	0:12:00
58	0:11:40	0:12:08	0:12:13
59	0:11:48	0:12:15	0:12:18
60	0:11:51	0:12:46	0:13:00
61	0:11:54	0:12:49	0:13:01
62	0:12:32	0:13:02	0:13:04
63	0:12:42	0:13:08	0:13:14
64	0:12:50	0:13:34	0:13:36
65	0:12:55	0:13:48	0:14:02
66	0:13:01	0:14:03	0:14:07
67	0:13:25	0:14:13	0:14:20
68	0:13:38	0:14:47	0:14:53
69	0:13:56	0:14:59	0:15:01
70	0:14:11	0:15:07	0:15:17
71	0:14:43	0:15:24	0:15:27
72	0:14:51	0:15:33	0:15:36
73	0:15:05	0:15:45	0:15:47

74	0:15:17	0:15:59	0:16:02
75	0:15:25	0:16:07	0:16:17
76	0:15:39	0:16:18	0:16:21
77	0:15:43	0:16:23	0:16:26
78	0:15:50	0:16:30	0:16:32
79	0:15:57	0:16:49	0:16:52
80	0:16:10	0:16:54	0:16:57
81	0:16:20	0:17:01	0:17:08
82	0:16:25	0:17:10	0:17:13
83	0:16:41	0:17:28	0:17:32
84	0:16:44	0:17:35	0:17:39
85	0:17:03	0:17:39	0:17:42
86	0:17:09	0:17:49	0:17:52
87	0:17:16	0:17:53	0:17:57
88	0:17:18	0:18:18	0:18:26
89	0:17:23	0:18:27	0:18:30
90	0:17:27	0:18:31	0:18:35
91	0:17:59	0:18:31	0:18:37
92	0:18:04	0:18:39	0:18:41
93	0:18:08	0:18:48	0:18:50
94	0:18:16	0:18:54	0:18:59