



Universidad Tecnológica de La Habana

José Antonio Echeverría

cujae

Facultad de Ingeniería Informática

Herramienta de simulación para evaluar configuraciones de señalización en redes viales

Informe de Prácticas Profesionales de 4to año

Autor: Ariadna Claudia Moreno Román (amoreno@ceis.cujae.edu.cu)

Tutor: Dr. C. Mailyn Moreno Espino (my@ceis.cujae.edu.cu)
Profesora Titular de la Facultad de Ingeniería Informática
Cujae, La Habana, Cuba.

La Habana, Cuba

Enero, 2021

Resumen

La mala planificación del tránsito puede traer consigo efectos no deseados como los embotellamientos, los colapsos y los accidentes viales. En estos sucesos, además de perderse vidas humanas y recursos materiales, es contaminado el medio ambiente a través del humo de los vehículos; y también de manera sonora por el sonido de los *clap-sons*. Para evitar situaciones como esta es necesario, antes de decidir colocar señalizaciones viales, saber si las mismas tendrán el resultado esperado o saber qué situaciones provoca.

El presente trabajo describe el proceso de desarrollo de una herramienta de simulación basada en agentes que contribuye al proceso de configuración de controladores del tránsito y otras señalizaciones. La solución está fundamentada en una investigación previa cuyo resultado es una herramienta que usa información de mapas geolocalizados para la simulación basada en agentes utilizando el marco de trabajo MASON. La misma era capaz de simular la vía, con algunos de sus componentes básicos, para evaluar la puesta en marcha de un semáforo y su configuración. A esta aplicación se le incorporan comportamientos que contemplen la sensación térmica del ambiente, las características de los vehículos y su influencia en la magnitud de los accidentes, el estado físico del pavimento de las calles, la presencia de animales y peatones, la experticia de los conductores y nuevas señales viales, lo cual busca que la aplicación se acerque cada vez más a la vida vial.

Con el desarrollo de la herramienta, analistas e ingenieros de tránsito cuentan con un sistema de software que les aporta información útil al análisis y toma de decisiones en cuanto a los problemas relacionados con la configuración de señales del tránsito.

Palabras claves: Embotellamiento, Semáforo, Señal de tráfico, Simulación Basada en Multiagente

Abstract

The bad planning of the traffic can bring not wanted effects as the jams, the collapses and the traffic accidents. In these events, besides getting lost human lives and material resources, the environment is contaminated through the smoke of the vehicles; and also in a sound way for the sound of the clap-sons. To avoid situations like this it is necessary, before deciding to place traffic signaling, to know if the same ones will have the prospective result or to know what situations it causes.

The present work describes the process of development of an agent based simulation tool that contributes to the process of traffic's controllers configuration. This solution is based in a previous investigation whose result is a tool that uses information of geolocalized maps for the agent based simulation using the MASON framework. The same was able to simulate the road, with some of their basic components, to evaluate the setting in march of a traffic light and their configuration. To this application have been incorporated behaviors that contemplate the thermal sensation of the atmosphere, the characteristics of the vehicles and its influences in the magnitude of the accidents, and the physical state of the pavement of the streets, the witnesses of animals and pedestrians, the experience of the drivers and new traffic signs, which looks for the application comes closer more and more to the road life.

With the development of the tool, analysts and traffic engineers have a software system that contributes them useful information to the analysis and taking of decisions for those problems related with the configuration of signs of the traffic.

Key words: Jams, Multiagent Based Simulation, Traffic lights, Traffic signal

Índice General

Introducción	1
CAPÍTULO 1. Simulación basada en agentes y señalización vial	5
1.1. El fenómeno del tránsito	5
1.1.1. Señalización de las vías	5
A. Los semáforos	6
B. Señales verticales	8
C. Agentes de circulación	10
D. Señales horizontales	11
E. Consecuencias de la mala configuración de las señales en la vía	12
1.1.2. Factores que influyen en el comportamiento de la vía	14
A. Estado físico del pavimento de las carreteras	15
B. Sensación ambiental	15
C. Animales en la vía	16
D. Características de los vehículos	16
E. Vehículos que adelantan carril	18
F. Factor humano en la vía	18
1.1.3. Importancia de una correcta sincronización	19
1.2. Sistemas inteligentes y Simulación	19
1.2.1. Los agentes	19
1.2.2. Sistemas Multi-Agente	20
1.2.3. Modelado y simulación	20
1.2.4. Simulación Basada en Multi-Agente	21
1.2.5. Modelos matemáticos	22
1.2.6. Modelos basados en lógica difusa	23
1.3. Sistemas de Información Geográfica	24
1.4. Herramientas de configuración de señales existentes en el mundo	25
1.5. Conclusiones parciales	28
CAPÍTULO 2. Herramienta de Simulación de Tráfico	29
2.1. Modelado del negocio	29
2.1.1. Modelo del dominio	29
2.1.2. Reglas del Negocio	31
2.1.3. Diagrama de Actividades	32
2.2. Captura de Requisitos	33
2.2.1. Requisitos Funcionales	33

2.2.2. Requisitos No Funcionales	34
2.3. Evaluación de arquitecturas candidatas	35
2.3.1. Plataformas de agentes	35
2.3.2. MASON como plataforma de simulación	36
2.4. Casos de Uso del Sistema	41
2.4.1. Actores del Sistema	41
2.4.2. Diagrama de Casos de Uso del Sistema	41
2.4.3. Especificación de los Casos de Uso del Sistema	42
2.5. Vista de la arquitectura	43
2.5.1. Descripción de los paquetes	43
2.6. Clases Principales	44
2.6.1. Agentes de señalización	44
2.6.2. Agente Vehículo	45
2.6.3. Agente Conductor	46
2.6.4. Agentes que modelan los factores externos presentes en la simulación . . .	47
2.7. Patrones y principios de diseño	48
2.7.1. Patrón Singleton	48
2.7.2. Patrón Facade	49
2.7.3. Patrón Observer	51
2.7.4. Patrón DAO	51
2.8. Modelo de datos	52
2.8.1. Datos espaciales	53
2.8.2. Datos estadísticos	55
2.9. Interfaz de usuario	58
2.10. Conclusiones parciales	60
CAPÍTULO 3. Validación de la Herramienta	61
3.1. Pruebas Funcionales	61
3.1.1. Guiadas a casos de uso	61
3.1.2. Clases de equivalencia	63
3.1.3. Análisis de los resultados de las pruebas	64
3.2. Comparación de resultados con diferentes configuraciones	65
3.3. Diseño de Experimentos	67
3.3.1. Fase de planeación	67
3.3.2. Fase de diseño	68
3.3.3. Fase de conducción	68
3.3.4. Fase de análisis	69
3.3.5. Conclusiones del DoE	71
3.4. Conclusiones parciales	71
Conclusiones generales	72
Recomendaciones	73
Referencias bibliográficas	I

ANEXOS	V
A. Plataformas de Agentes	VI
A.1. MASON	VI
A.2. JADE	VII
A.3. JADEX	VIII
A.4. NetLogo	IX
A.5. RePast	XI
A.6. GAMA	XII

Índice de Figuras

Figura 1.	Semáforos de tres secciones	6
Figura 2.	Semáforos con secciones adicionales y flechas	7
Figura 3.	Señales de prioridad	9
Figura 4.	Señales modificadoras de velocidad	9
Figura 5.	Proximidad de un reductor de velocidad	10
Figura 6.	Señales horizontales transversales a la circulación	12
Figura 7.	Principales causas de muerte en Cuba	13
Figura 8.	Reporte de accidentes por causas en Cuba	14
Figura 9.	Tipos de vehículos	17
Figura 10.	Comportamiento de un agente inteligente	20
Figura 11.	Esquema general del proceso de modelado científico	21
Figura 12.	Modelo de dominio	30
Figura 13.	Diagrama de actividades	32
Figura 14.	Arquitectura MASON, Patrón Repositorio	36
Figura 15.	Arquitectura MASON, Patrón MVC	37
Figura 16.	Arquitectura MASON, Patrón Flujo de Datos	37
Figura 17.	Relaciones primarias entre el modelo y la GUI visualización / control . . .	38
Figura 18.	Diagrama de la simulación de núcleo de MASON y de la programación de eventos discretos	39
Figura 19.	Diagrama del paquete Network de MASON	39
Figura 20.	Diagrama del código de control de GUI de MASON	40
Figura 21.	Instalación Inspector de MASON	40
Figura 22.	Diagrama de Casos de Uso del Sistema	42
Figura 23.	Diagrama de estructuración en capas. Enfoque por reutilización	43
Figura 24.	Diagrama de Clases relacionadas con el Agente Semáforo	45
Figura 25.	Diagrama de Clases relacionadas con el Agente Vehículo	46
Figura 26.	Diagrama de Clases relacionadas con el Agente Conductor	46
Figura 27.	Diagrama de Clases relacionadas con el Agente Pavimento	47
Figura 28.	Diagrama de Clases relacionadas con el Agente Ambiente	48
Figura 29.	Patrón <i>Sinlgeton</i> , Clase TrafficSimGeo	49
Figura 30.	Diseño de las utilidades, iteración 1	49
Figura 31.	Diseño de las utilidades, iteración 2 (Patrón <i>Facade</i>)	50
Figura 32.	Patrón Observer	51
Figura 33.	Patrón DAO	52
Figura 34.	Modelo lógico de la Base de Datos Espacial	53
Figura 35.	Modelo físico de las tablas	54

Figura 36.	Modelo lógico de las tablas	55
Figura 37.	Modelo físico de las tablas	56
Figura 38.	Interfaz de usuario	58
Figura 39.	Notificaciones del sistema	59
Figura 40.	Interfaz de usuario	59
Figura 41.	Variantes para obtener mapas	60
Figura 42.	Accidentes, infracciones y roturas	66
Figura 43.	Velocidad promedio de los vehículos	66
Figura 44.	Tiempo promedio en espera	67
Figura 45.	Gráfica de efectos principales	69
Figura 46.	Gráfica de Pareto de efecto y rendimiento	69
Figura 47.	Gráfica de probabilidad normal	70
Figura 48.	Gráfica de cubos	70

Índice de Tablas

Tabla 1.	Posibilidad de accidentes según color y momento del día	16
Tabla 2.	Velocidades máximas según tipo de vehículo	17
Tabla 3.	Definición de los conceptos del Modelo del Dominio (1)	30
Tabla 4.	Definición de los conceptos del Modelo del Dominio (2)	31
Tabla 5.	Identificación de requisitos	33
Tabla 6.	Requisitos No Funcionales (1)	34
Tabla 7.	Requisitos No Funcionales (2)	35
Tabla 8.	Comparación de las plataformas de agentes	36
Tabla 9.	Descripción de los Actores del Sistema	41
Tabla 10.	Descripción de alto nivel del Caso de Uso “Configurar simulación”	42
Tabla 11.	Descripción de alto nivel del Caso de Uso “Ejecutar simulación”	42
Tabla 12.	Descripción de alto nivel del Caso de Uso “Exportar datos”	42
Tabla 13.	Descripción de los paquetes de la arquitectura	44
Tabla 14.	Descripción de los atributos de la tabla “roads”	54
Tabla 15.	Descripción de los atributos de la tabla “roads nodes”	54
Tabla 16.	Descripción de los atributos de la tabla “stats” (1)	56
Tabla 17.	Descripción de los atributos de la tabla “stats” (2)	57
Tabla 18.	Descripción de los atributos de la tabla “conductor”	57
Tabla 19.	Descripción de los atributos de la tabla “signal”	57
Tabla 20.	Descripción de los atributos de la tabla “accident”	58
Tabla 21.	Caso de Prueba: Cargar pronóstico del tiempo del servicio web	62
Tabla 22.	Caso de Prueba: Identificar vehículos infractores y afectados	62
Tabla 23.	Caso de Prueba: Detener simulación	63
Tabla 24.	Clases de equivalencia	63
Tabla 25.	Prueba de las clases de equivalencia	64
Tabla 26.	Juegos de datos de las ejecuciones	65
Tabla 27.	Parámetros comunes para todas las simulaciones	65
Tabla 28.	Factores controlables, niveles y unidades de medidas	68
Tabla 29.	Rendimiento de cada tratamiento	68
Tabla 30.	Configuración Recomendada	71

Introducción

En la actualidad, la puesta en marcha de un proceso o sistema sin antes tener cierto grado de confianza de cómo será su funcionamiento, no constituye una buena práctica. Esto es provocado por el riesgo de ocurrencia de colapsos, insuficiencias y errores, es decir, obtención de resultados no esperados. Las técnicas de simulación se han utilizado durante mucho tiempo de una forma eficaz para imitar el desempeño de un sistema o artefacto que se desea desarrollar, con el fin de estimar cuál sería su desempeño e impacto real [1].

Existen varios tipos de simulación computacional, entre ellas, no se puede determinar cuál es la más conveniente utilizar para todos los tipos de problemas [2, 3]. Según las características de un modelo, el tipo de simulación a aplicar, puede brindar resultados cercanos a los reales o no.

La Simulación Basada en Multi-Agente (MABS, *Multi-Agent Based Simulation*) se usa actualmente en un número creciente de áreas. Esto se debe, en su mayor parte, a su capacidad para hacer frente a una amplia variedad de modelos, que van desde entidades simples, generalmente llamadas “agentes reactivos”, a otras más complejas, como los “agentes cognitivos” [4].

Una de las temáticas que puede abordar la MABS es la ingeniería de tránsito. En Cuba actualmente la colocación de semáforos en intersecciones de carreteras se realiza sin tener un previo conocimiento del impacto que tendrá. La instalación de estas señalizaciones supone para el país una inversión de recursos materiales y mano de obra que se desperdician si el resultado es contraproducente. Además, la mala configuración semafórica trae como consecuencia embotellamientos en las vías, accidentes y contaminación, y por supuesto pérdidas económicas.

Un ejemplo de mala configuración semafórica se instanció en octubre de 2015, en la Rotonda de la Ciudad Deportiva, La Habana, Cuba. El proyecto consistió en instalar un semáforo en cada uno de los accesos a la rotonda, regulando así la entrada a la misma [5]. El impacto que tuvo esta reforma fue una congestión en las vías adyacentes por más de 25 minutos, automóviles atascados encima del paso a nivel ferroviario e interrupción en la entrada del Hospital Clínico Quirúrgico de 26. Conclusión, la Dirección Nacional de Tránsito indica retirar los semáforos en menos de 72 horas de su puesta en funcionamiento. Evidentemente, este procedimiento “pone en tela de juicio la calidad del estudio previamente anunciado” [6].

El presente trabajo tiene como antecedente la Tesis de Pregrado del Ing. Dayan Bravo Fraga [7]. En la misma se modelan como agentes los vehículos y los semáforos como componentes principales, y a esta herramienta fue añadida la posibilidad de que los vehículos puedan adelantarse por el carril en donde están transitando y la funcionalidad de la luz amarilla, pero sigue teniendo como limitantes que no contempla diferencias entre el tamaño de los vehículos, ni se modifica el comportamiento de los mismos según la temperatura del ambiente ni por las características del pavimento.

Partiendo de la **situación problemática** anteriormente descrita, es posible plantear el **problema a resolver**:

¿Cómo puede la Simulación Basada en Agentes apoyar la toma de decisiones en la implementación de las señalizaciones de tránsito?

El problema antes expuesto está abarcado por el siguiente **objeto de estudio**: la Inteligencia Artificial Distribuida (DAI), Simulación Computacional, Ingeniería del Tránsito y los Sistema de Información Geográfica (GIS); y de este amplio conjunto centra la atención en el **campo de acción** que comprenden: el comportamiento vial, la MABS, los datos espaciales y la señalización de las redes viales.

Para dar solución al problema se plantea el siguiente **objetivo general**:

Incluir a la Herramienta de Simulación para evaluar configuraciones de señalización en redes viales la posibilidad de que las características del vehículo, del ambiente, del pavimento, el efecto de otro tipo de señalizaciones, la experticia de los conductores y la presencia de peatones, animales y otros obstáculos en la vía influyan sobre el comportamiento de los mismos.

Como **objetivos específicos** se tienen los siguientes, cada uno desglosado con las **tareas a desarrollar**:

1. Describir los fundamentos teóricos que sirven de base al desarrollo de la herramienta.
 - Caracterizar los agentes inteligentes y los Sistemas Multi-Agente.
 - Asimilar los conceptos de la simulación y de la MABS.
 - Estudiar funcionamiento de los GIS.
 - Estudiar las bases de los modelos basados en lógica difusa.
 - Estudiar Ley 109 código de vialidad y tránsito, Capítulo 3 “de la Señalización”.
 - Asimilar todo el modelo de Simulación desarrollado para la evaluación semafórica.
 - Realizar análisis estadístico de las características reales de los vehículos en La Habana.
2. Incorporar nuevas funcionalidades a la herramienta de simulación para que los vehículos actúen según sus características y las de su conductor, perciban y respondan a factores externos y obedezcan nuevas señales.
 - Adecuar comportamiento de los vehículos para que contemplen su estado técnico y que puedan ocurrir desperfectos en plena marcha.
 - Adecuar comportamiento de los vehículos para que actúen según su tamaño.
 - Adecuar probabilidad de los accidentes según la distribución por colores de los vehículos.
 - Registrar impacto de los accidentes según el tamaño de los vehículos relacionados.

- Adecuar comportamientos de los vehículos para que actúen según la temperatura, humedad y lluvia en el ambiente.
 - Agregar a las carreteras un estado físico.
 - Adecuar comportamientos de los vehículos para que moderen su velocidad según el estado del pavimento.
 - Adecuar comportamientos de los vehículos para que eviten colisionar con los animales y los peatones (obstáculos).
 - Agregar nuevas señales a la simulación.
 - Adecuar comportamientos de los vehículos para que actúen según las nuevas señales colocadas.
 - Utilizar lógica difusa para obtener grupos de conductores según su experticia y edad.
 - Adecuar comportamientos de los vehículos para que actúen según la experticia y edad de su conductor.
3. Construir modelo matemático inicial para la simulación.
- Identificar variables de decisión y parámetros.
 - Identificar función objetivo y restricciones del modelo.
 - Construir modelo matemático.
4. Validar la herramienta desarrollada.
- Desarrollar pruebas funcionales.
 - Probar el software con un número elevado de vehículos circulando para verificar que sea capaz de imitar el flujo real de autos en la vía.
 - Realizar pruebas de compatibilidad.

El **aporte** de este proyecto está dado tanto en el marco teórico como en el práctico. Para el primero, se construye un modelo de simulación semafórica cuya salida es un software de sistema extensible y útil para la solución de cualquier problema en redes viales. En la práctica, queda un sistema de software para el apoyo en la toma de decisiones que ahorrará tiempo y esfuerzo a los analistas que hagan estos estudios.

En cuanto a la **estructuración**, este trabajo está dividido en tres capítulos. Primeramente, en el **Capítulo 1: Simulación basada en agentes y señalización vial**, está el resultado del estudio realizado sobre las temáticas que aborda el trabajo, el estado del arte de la vía en la actualidad, se caracterizan los agentes inteligentes y los sistemas multi-agente, se explica en que consiste la simulación computacional, así como los sistemas de información geográficos y se realiza un estudio de herramientas similares desarrolladas en el mundo. El **Capítulo 2: Herramienta de Simulación de Tráfico**, contiene la documentación sobre el desarrollo de

la herramienta, los artefactos de Ingeniería de Software construidos, la captura de requisitos para el software, las consideraciones tomadas para la interfaz de usuario, el mecanismo de acceso a datos, los patrones de diseño utilizados y cómo fueron implementadas las principales clases de la herramienta. Finalmente, el **Capítulo 3: Validación de la Herramienta**, presenta la documentación de las pruebas que se le realizaron al software, tanto las funcionales guiadas por casos de prueba como las clases de equivalencia. También se hace una comparación entre los resultados arrojados por la herramienta para diferentes configuraciones, y termina el capítulo con un Diseño de Experimentos para el caso de estudio de la Ciudad Deportiva.

CAPÍTULO 1

Simulación basada en agentes y señalización vial

En el presente capítulo se encuentra lo referente a las bases teóricas de la herramienta, que comprende un análisis del proceso actual de colocación de controladores de tránsito, las consecuencias de la mala configuración y la importancia de una correcta sincronización. Se muestra el impacto de los factores que influyen en la velocidad y el comportamiento de los vehículos en la vía. También se explica el paradigma orientado a agentes y los Sistemas Multi-Agente, dejando referenciadas las características peculiares de estos. Se explica en qué consiste la simulación computacional, resaltando la Simulación Basada en Agentes y cómo pudiese la misma favorecer en este procedimiento y también se explica qué son los modelos matemáticos y cómo apoyan a la simulación. Se introduce el término de Lógica Difusa y cómo se pudiera utilizar para enriquecer la herramienta a desarrollar. Además, se exponen las bases de los Sistema de Información Geográfica y su aplicación. Por último, se presenta un estudio sobre herramientas de simulación similares desarrolladas en el mundo en los últimos años, sus limitaciones y qué beneficios se pueden obtener de ellas.

1.1. El fenómeno del tránsito

En el último siglo el transporte se ha convertido en una parte fundamental de la vida citadina. Como resultado de ello, se han incrementado de manera considerable el número de usuarios que transitan por las vías en vehículos de cualquier tipo. Con este aumento, se incrementa también lamentablemente, el número de efectos negativos, como son los embotellamientos, colapsos, contaminación y accidentes. Una de las causas que agrava esta situación es la mala configuración de controladores del tráfico y otras señales viales.

1.1.1. Señalización de las vías

Las señales de tráfico o señales de tránsito son los signos usados en la vía pública para impartir la información necesaria a los usuarios que transitan por un camino o carretera, en especial los conductores de vehículos y peatones [8].

Son las siguientes según su prioridad:

- Señales y órdenes de los agentes de circulación
- Señalización circunstancial que modifique el régimen normal de utilización de la vía y señales de balizamiento fijo
- Señales mediante luces de los semáforos, sonoras y lumínicas.
- Señales verticales de circulación
- Señales horizontales

Si se diera el caso de que se encuentran varias señales contradictorias, prevalecerá la más prioritaria, y en el caso de que sean del mismo grupo, la más restrictiva [8]. Puesto que las diferencias de idioma pueden crear barreras, las muestras internacionales usan símbolos en lugar de palabras. Se han desarrollado principalmente en Europa y se han adoptado en la mayoría de los países, incluyendo a Cuba.

A. Los semáforos

Las señales mediante luces se utilizan para regular la circulación vial y generalmente se observan mediante semáforos, divididos los más comunes en tres secciones, verticales u horizontales, situándose sus luces de arriba abajo o de izquierda a derecha respectivamente, siempre manteniendo el siguiente orden: roja, amarilla y verde como muestra la Figura 1. Además, existen otros tipos de semáforos como son los semáforos con direcciones, los semáforos de ocupación de carril y los semáforos de peatones [9].

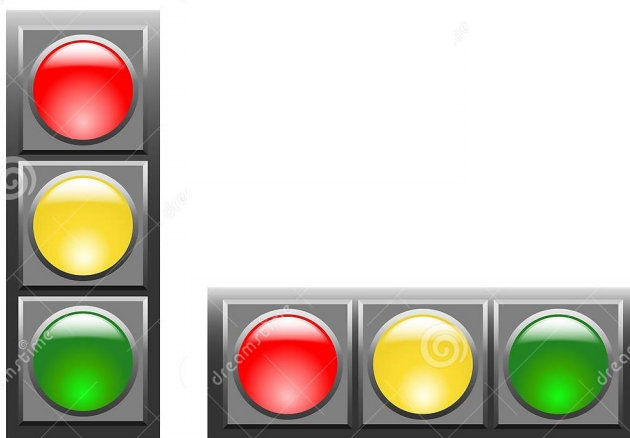


FIGURA 1: Semáforos de tres secciones

Los semáforos pueden disponer de un contador que permite a los conductores conocer el tiempo restante de cada luz, que no sustituye sus significados, ni determina las prioridades de las corrientes vehiculares.

Luz roja: indica que los conductores de vehículos deben detener la marcha en la línea de pare, de existir, o en la línea de paso para peatones, en las inmediaciones de la intersección sin sobrepasar la misma o en la vertical de la señal correspondiente si la hubiese, y los peatones, en la acera o zona de seguridad.

Luz amarilla: indica que los conductores de vehículos deben detener la marcha en la línea de pare y de no existir, hacerlo en el paso para peatones, a no ser que, cuando se proyecte la luz amarilla, se encuentre tan cerca de la intersección que no pueda detener el vehículo en condiciones de seguridad suficientes; pudiera parecer que se trata de la misma definición, pero no es así, la luz amarilla resulta tanto para conductores como para peatones una señal de aviso, del paso tanto de la luz verde para roja como viceversa; es una luz de aviso o preventiva.

Luz verde: indica a los conductores de vehículos que pueden continuar la marcha por la misma vía que circulan, o girar a la derecha o izquierda según el caso. Se excepcionan de esta regulación los carriles para los que se disponga de una fase semafórica para el giro de izquierda, simultánea o no con el sentido recto.

Luz verde con flecha: solo se permite a los conductores de los vehículos continuar la marcha en el sentido que indique la flecha, independientemente del sentido de circulación de la vía, se muestran ejemplos en la Figura 2.

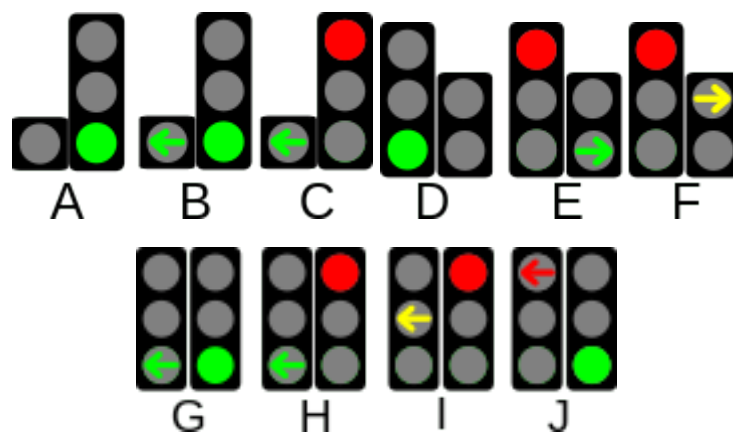


FIGURA 2: Semáforos con secciones adicionales y flechas

Secciones adicionales: en los ejemplos de la Figura 2 los conductores pueden continuar la marcha en el sentido en que indican las flechas con independencia de la luz proyectada por el sistema tricolor [9].

La **configuración** de los semáforos consiste en determinar las fases que conformarán su ciclo, así como el tiempo que tendrá cada una de ellas. También se contempla, como una alternativa de configuración, la ausencia de estos [7].

B. Señales verticales

Se denomina señal vertical al conjunto compuesto por elementos de sustentación, placa y símbolo o leyenda específica inscritos en ella [9]. Esas señales persiguen como objetivos la seguridad, eficiencia y comodidad de la circulación. Para ello advierte los posibles peligros, ordena la circulación de acuerdo con las circunstancias locales, recuerda algunas prescripciones del Código que proporciona al usuario una información conveniente.

Las señales por su contenido y su significado, se dividen en los principales grupos siguientes [9]:

- Grupo A: Señales de peligro o precaución
- Grupo B: Señales de prioridad
- Grupo C: Señales de prohibición
- Grupo D: Señales de obligación
- Grupo E: Señales de fin de prohibición u obligación
- Grupo F: Señales de información
- Grupo G: Señales de orientación
- Grupo H: Señales para los pasos a nivel

Entre las señales de prioridad existen algunas que modifican directamente el comportamiento de los vehículos ante una determinada intersección de carreteras, las mismas se detallan a continuación y se muestran en la Figura 3.

Pare: indica que todos los vehículos están obligados a detenerse en la línea de “Pare” y si no la hubiera, antes del borde de la línea transversal sin detenerse en ella, cualesquiera que sean las circunstancias de visibilidad, dándole prioridad a los vehículos que circulan por la vía transversal antes de cruzarla o incorporarse a ella [9].

Ceda el paso: indica que todo vehículo está obligado a disminuir la velocidad y parar si fuera necesario, para permitir el paso a todos los vehículos que circulan por la vía preferente [9].

Ceda el paso con luz roja: indica a los conductores de vehículos que pueden realizar giros de derecha o seguir recto en las intersecciones semaforizadas, con extrema precaución, cediéndole

el paso a los conductores y peatones que transitan por la intersección con la luz verde correspondiente y a los peatones que cruzan la vía transversal con la luz verde de peatones [9].

Ceda el paso izquierda con luz verde: indica a los conductores de vehículos que pueden realizar giros de izquierda, con extrema precaución, cediéndole el paso a los vehículos que circulan en sentido opuesto en intersecciones semaforizadas donde está permitida esta maniobra, una vez que haya cesado la fase que establece el giro y mientras se mantenga la luz verde [9].

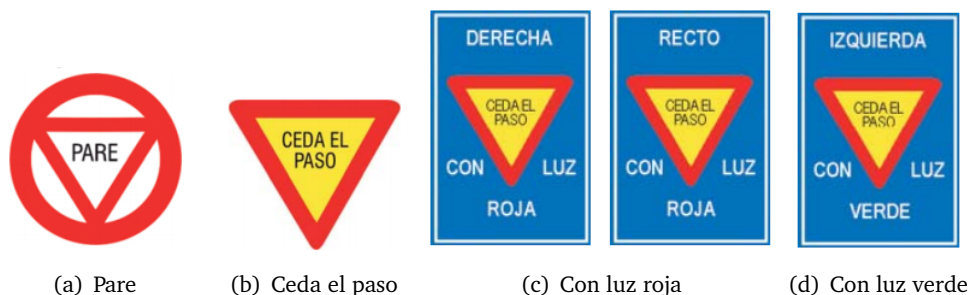


FIGURA 3: Señales de prioridad

La velocidad de los vehículos también es controlada mediante señales verticales, pertenecientes a varios de los grupos, que indican máximos y mínimos de velocidad para un tramo determinado de carretera, así como cuál sería la velocidad más apropiada para transitarla. Estas señales se ilustran en la Figura 4.

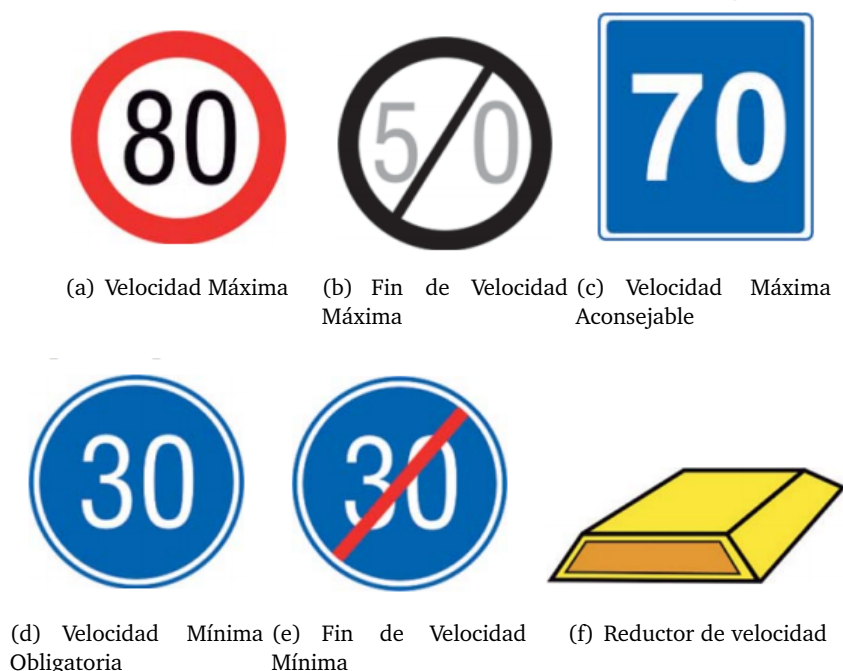


FIGURA 4: Señales modificadoras de velocidad

Velocidad Máxima: (Grupo A) establece el límite máximo de velocidad al que se puede circular.

Fin de restricción de límite máximo de velocidad: (Grupo E) indica el cese de la restricción del límite de velocidad establecido para un tramo de la vía.

Velocidad Mínima Obligatoria: (Grupo D) indica que los vehículos que circulan por la vía donde está colocada la señal no pueden circular a una velocidad menor que la que está indicada.

Fin de velocidad mínima obligatoria: (Grupo E) indica que deja de ser obligatoria la velocidad mínima establecida.

Velocidad máxima aconsejable: (Grupo G) indica la conveniencia de no rebasar la velocidad señalada [9].

Reductor de velocidad: no pertenece a ningún grupo pues no es una señal vertical en sí, es una brusca variación que sobresale del pavimento y lo atraviesa de lado a lado, para inducir a los conductores a reducir la velocidad de su vehículo. Generalmente, por el peligro que supone, está avisado previamente por la correspondiente señal de tráfico mostrada en la Figura 5, acompañada de una limitación de velocidad que desaparece después del peligro anunciado. Si un vehículo pasa demasiado rápido sobre un reductor, se puede dañar el sistema de suspensión, o incluso otras partes, dependiendo del impacto [10].



FIGURA 5: Proximidad de un reductor de velocidad

De manera general, las señales verticales juegan un importante papel en la circulación de los vehículos. En intersecciones donde el flujo de tráfico no demanda una señal semafórica es imprescindible tener en cuenta el posicionamiento de señales verticales con el fin de regular los cruces y evitar accidentes.

C. Agentes de circulación

Algunas de las señales que realiza el agente de la autoridad en la vía son [9]:

Atención, alto: todos los usuarios de la vía deben detenerse de inmediato.

Alto: todos los usuarios de la vía que se encuentre de frente o a su espalda deben detenerse de inmediato y solamente los que se encuentran en igual dirección del brazo que levanta pueden continuar en el mismo sentido o realizar todas aquellas maniobras que no están prohibidas en el cruce o intersección.

Brazo extendido horizontal al frente: vehículos a la izquierda pueden circular en todas las direcciones y los peatones pueden cruzar a su espalda.

Balanceo manual de luz roja: para los usuarios de la vía hacia los cuales está dirigida la luz, que deben detenerse.

Además, puede auxiliarse de un silbato con una serie de toques cortos para detener los vehículos y un toque largo para reanudar la marcha; de un bastón lumínico al hacer regulaciones de brazos señaladas, impartir órdenes e instrucciones para la mejor regulación de la circulación, y de las manos para la reducción o aumento de la velocidad, parar o seguir [9]. Los agentes realizan otro conjunto de señales, pero solo se documentan las que sirven para la herramienta que se desarrolla.

D. Señales horizontales

Las señales horizontales son aquellas marcas viales que se hacen sobre el pavimento y se emplean para regular o encauzar la circulación, advertir e informar a los usuarios de la vía.

El empleo de las señales horizontales se rige por las normas generales siguientes [9]:

- Se utilizan solas o con otros medios de señalización, a fin de reafirmar o precisar las indicaciones.
- Son amarillas o blancas, pero su color no indica la regulación, sino el diseño de su trazado en el pavimento.
- Deben cumplir con los parámetros técnicos adecuados de visibilidad diurna, retroflexión y resistencia al deslizamiento y a la intemperie.

Estas señales de acuerdo con su objetivo se clasifican en [9]:

Marcas paralelas a la circulación: dividen los carriles o sendas y regulan los tramos donde su permite o prohíbe el adelantamiento o el pase a otro carril o senda.

Marcas transversales a la circulación: dispuestas de forma perpendicular al sentido de la circulación de las corrientes vehiculares, pueden indicar “Pare”, “Ceda el paso”, entre otros.

Otras marcas: varían desde prohibir la circulación sobre ellas, hasta indicar la posibilidad de giros o proximidad de obstáculos.

De todas las señales horizontales, las marcas transversales a la circulación son las que más contribuyen a contrarrestar los accidentes pues apoyan a las señales verticales o semafóricas en la regulación de los cruces. Algunas de ellas se explican a continuación [9], ilustrándolas en la Figura 6.

Línea de Pare: indica que el conductor está obligado a detenerse ante ella por estar ante la señal de “Pare” o la señal de luz amarilla o roja del semáforo.

Señal de Pare: Las líneas de “Pare” pueden llevar delante escrita sobre la calzada la palabra “Pare” tantas veces como se entienda necesario.

Ceda el Paso: indica que los conductores no deben normalmente pasarla cuando tienen que ceder el paso por estar instalada la señal de ese nombre.

Paso Peatonal: indica el lugar destinado para que los peatones crucen la vía mediante la marca tipo cebra con preferencia en el cruce sobre los vehículos.

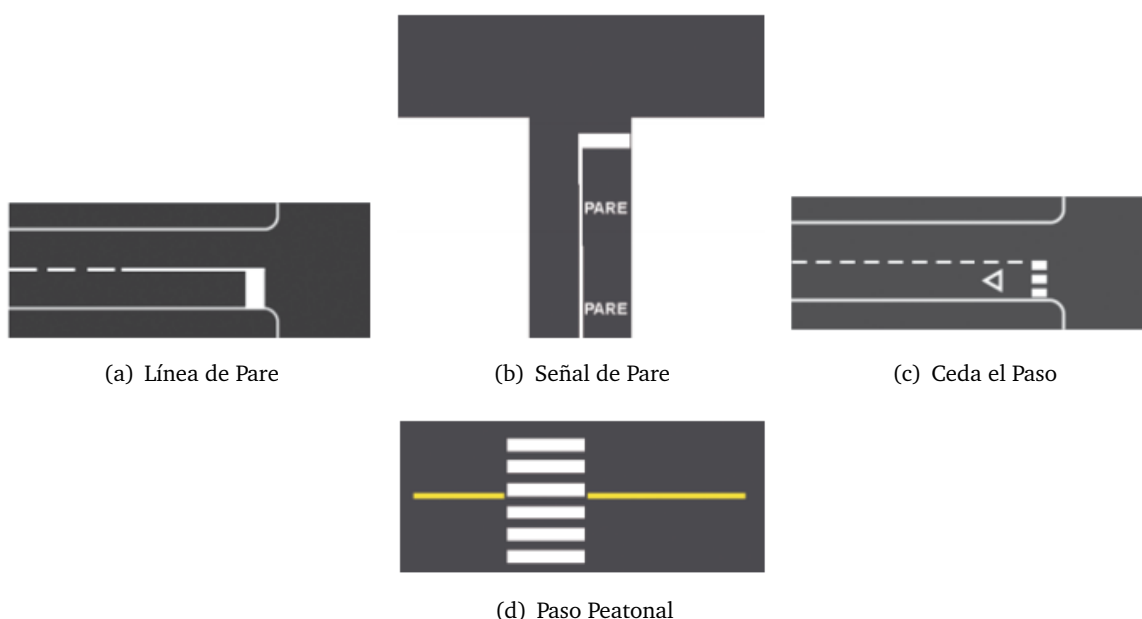


FIGURA 6: Señales horizontales transversales a la circulación

Las señales horizontales suelen ser pasadas por alto con más frecuencia debido a que su pintura se desgasta con el paso del tiempo, es responsabilidad de las autoridades del tránsito mantenerlas visibles para que los conductores no cometan infracciones sobre las mismas.

E. Consecuencias de la mala configuración de las señales en la vía

Las malas configuraciones de las señales en la vía son aquellas que, lejos de organizar el transporte, lo obstaculizan. Se realizan sin tener certeza del impacto que tendrán en la práctica. Cuando se pone en funcionamiento una mala configuración de cierta señalización vial, las afectaciones ocurrientes son proporcionales a la demanda de vehículos que tenga la carretera en cuestión. Las consecuencias que traen las mismas pueden provocar desde una molesta contaminación sonora, hasta acabar con la vida de un ser humano.

Embotellamientos

Los embotellamientos o colapsos del tránsito son aglomeraciones excesivas de vehículos en espera por la luz del semáforo o por otros vehículos. Ellos constituyen una de las principales afectaciones que genera la mala configuración de los semáforos. Una situación como esta fue instanciada en octubre de 2015, en la Ciudad Deportiva, La Habana, Cuba. En esa fecha se aprobó y puso en ejecución un proyecto que consistía en instalar un semáforo en cada acceso a la rotonda de la Ciudad Deportiva [5]. Como es evidente, se requirió hacer uso de recursos económicos y humanos. Este proyecto fue conocido por su impacto totalmente contraproducente. Provocó una enorme congestión en las vías por más de 25 minutos dejando automóviles varados encima del paso a nivel ferroviario y obstaculizaban la entrada del Hospital Clínico Quirúrgico de 26. La decisión de las autoridades relacionadas fue inhabilitar los semáforos cuando habían pasado apenas 72 horas de su puesta en funcionamiento [6].

Contaminación Ambiental

En principio, pudiese parecer absurdo que aplicar malas configuraciones en los controladores del tránsito contribuya a la contaminación ambiental. Pero una característica típica en los embotellamientos es que los vehículos avanzan con muy baja velocidad y con frecuentes paradas. Esto implica que los motores funcionen mucho más y, por lo tanto, generen más gases contaminantes que acaban en la atmósfera, contaminan el medio ambiente y fortalecen el cada vez más preocupante, calentamiento global. En Cuba, esta contaminación además se ve incrementada porque aún muchos de los vehículos tienen más de diez años, vehículos que por su antigüedad consumen más energía, producen mayor contaminación y son más ruidosos (contaminación acústica) [7].

Accidentes de tránsito

Los accidentes de manera general, constituyen la quinta causa de muerte en Cuba como muestra la Figura 7, y representan alrededor del 5.9% del total de fallecidos [11].

PRINCIPALES CAUSAS DE MUERTE EN CUBA

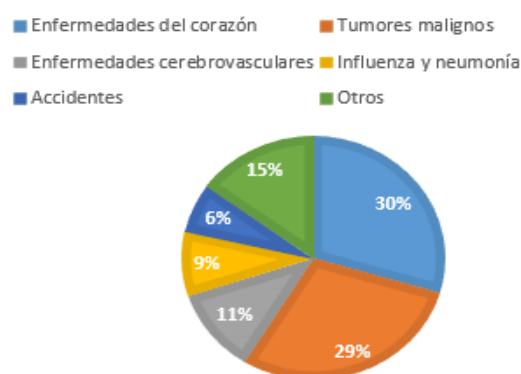


FIGURA 7: Principales causas de muerte en Cuba

Por esta causa se reportan 5.2 años de vida potencialmente perdidos por cada 1000 habitantes. De todos los accidentes que ocurren anualmente en Cuba, una gran parte están relacionados con la desobediencia de los conductores respecto a los semáforos. Estos datos se pueden corroborar gracias a la Oficina Nacional de Estadísticas e Información de la República de Cuba, ONEI [12].

La Figura 8 es un reporte por causas de los accidentes en Cuba, donde se muestran las causas más importantes, en relación con las cantidades de los años 2017 y 2018. Aquí se puede apreciar que no respetar el semáforo y demás señales en la vía, los excesos de velocidad y la distracción de los conductores constituyen unas de las más frecuentes causas de accidentes. La mayor parte de estas causas de accidentes serán modeladas durante la herramienta de simulación a realizar.

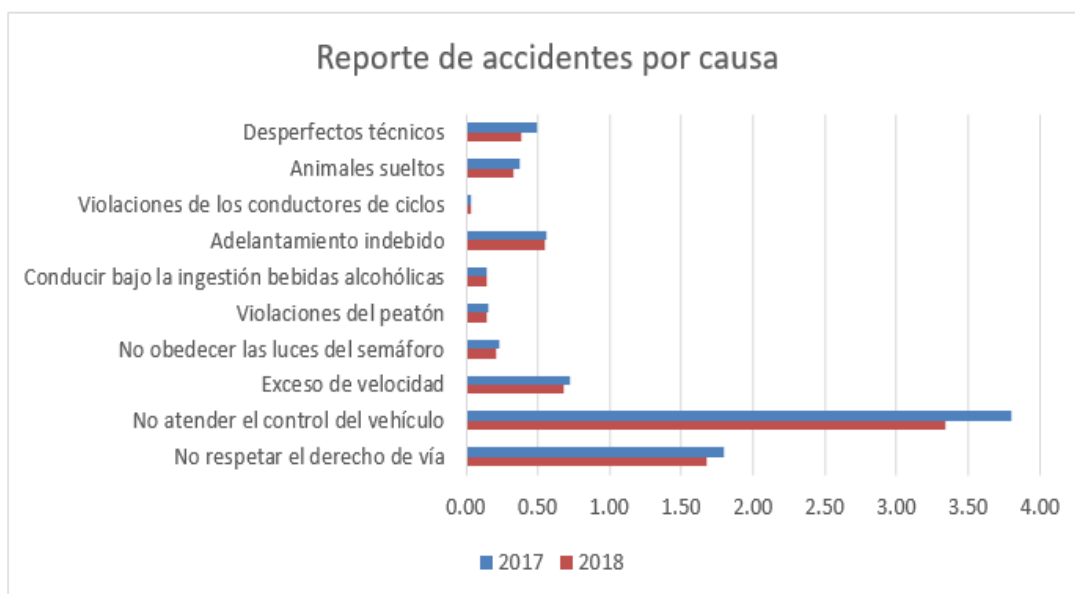


FIGURA 8: Reporte de accidentes por causas en Cuba

1.1.2. Factores que influyen en el comportamiento de la vía

El comportamiento de los vehículos en la vía está condicionado por varios factores que se dividen en tres categorías: humanos, del vehículo y externos. Las características y el estado técnico de los vehículos son determinantes a la hora de transitar pues de ello depende la velocidad y buena ejecución de las maniobras del mismo, y en caso de accidentes el impacto que tendrá. Los conductores se ven afectados por la temperatura, aumentando su velocidad en caso de sentirse incómodos e impacientes por llegar a su destino, además de otros factores psicológicos o situaciones que se le presenten e influyan en su buen conducir. El estado del pavimento es otro factor clave para que los vehículos moderen su velocidad en aras de cuidar el estado técnico del mismo. Y, por último, la vía puede verse interrumpida por peatones o animales que dificulten la circulación, poniendo en peligro su vida y la de los conductores.

A. Estado físico del pavimento de las carreteras

El pavimento es la capa constituida por uno o varios materiales que se coloca sobre el terreno natural o nivelado, para aumentar su resistencia y servir para la circulación de personas o vehículos [13, 14].

Como parte principal de la vía el estado físico del pavimento es decisivo en el comportamiento de los vehículos que por él transitan. El deterioro del mismo, especialmente la presencia de baches, genera crecientes restricciones de capacidad y aumenta la congestión. La lluvia acumulada sobre las calzadas reduce la capacidad de las vías y también genera congestión [15]. Por otra parte, las carreteras desgastadas se vuelven peligrosas a pesar de no tener baches, pues la lluvia provoca deslizamientos a los vehículos, que por ende, colisionan.

Los conductores cubanos, familiarizados ya con el estado de las carreteras, ante los desperfectos del pavimento disminuyen su velocidad, en ocasiones de manera brusca, provocando también accidentes por vehículos que los seguían en el mismo carril.

B. Sensación ambiental

Existen características adversas enmarcadas a condiciones de la vía, que pueden aumentar el nivel de exigencia del conductor, así como el riesgo en diferentes acciones, sin embargo, adicional a las características geométricas de la zona de circulación, se encuentran las condiciones climáticas que, desde leves aumentos en la temperatura, golpes de vientos fuertes y lluvia, pueden alterar la normal conducción y conllevar fácilmente a la ocurrencia de un siniestro en la vía [16, 17].

Con la lluvia las condiciones de visibilidad disminuyen, los parabrisas se empañan internamente y dificulta la detección de vehículos y personas en la vía. La visibilidad de los conductores también puede verse afectada por la humedad y la neblina. Así mismo, los neumáticos pierden adherencia y la rueda tiende a patinar sobre el agua con poco contacto al pavimento [16].

Ante altos grados de temperatura, que se reflejan aún más cálidos en el interior de los vehículos, los conductores manifiestan conductas de desesperación por arribar a su destino, desobedeciendo sobre la marcha las señales de tráfico que se les interpongan, y como es de suponer también aumentan su velocidad, siendo así más propensos a provocar o sufrir accidentes.

Además, también es determinante el instante del día que abarca la simulación a realizar. En horas de la mañana y la tarde los conductores suelen estar más atentos a lo que sucede en la vía; no obstante, en las noches y la madrugada, como el flujo de vehículos es menor los conductores se relajan y dejan de prestar atención a las señales, y de esta manera son causantes potenciales de accidentes.

C. Animales en la vía

Desgraciadamente existe un número no despreciable de animales que habitan en las calles, tratando de sobrevivir pasando frío, lluvias, hambre, y otras tantas miserias. Estos animales son parte de la vía e interrumpen la circulación de los vehículos frecuentemente. En ocasiones, por despreocupación o por no tener tiempo suficiente para maniobrar los vehículos terminan atropellándolos, quedando lastimados y en el peor de los casos muertos. Por parte de los conductores también supone un peligro pues al intentar salvar a estos animales realizan giros bruscos y apresurados que pueden terminar en un colapso o un accidente. Es un fenómeno común en la actualidad y merece ser tenido en cuenta en todos los estudios sobre tránsito.

D. Características de los vehículos

Las características de cada vehículo tienen mucho que decir sobre las probabilidades que tenga el mismo de colisionar. Un estudio realizado en la Universidad de Monash [18] establece una variación de la peligrosidad de los vehículos según su color y el momento del día, teniendo como partida la peligrosidad de los vehículos de color blanco.

La Tabla 1 presenta la relación entre los accidentes de tránsito y los distintos colores de automóviles definidos para Australia. Como unidad se utiliza la peligrosidad del color blanco. Por ejemplo, durante el día, el color plata (1,10) es 10 % más peligroso que el color blanco, al atardecer o al amanecer (1,15) es 15 % más peligroso y de noche (1,08) es 8 % más peligroso.

TABLA 1: Posibilidad de accidentes según color y momento del día

Posibilidad de accidentes en distintos momentos del día (blanco: 1.00)			
Color del vehículo	De día	Atardecer	De noche
blanco	1.01	1.00	1.05
anaranjado	0.09999	1.21	0.77
malva	1.07	1.00	0.65
beige	0.93	1.16	0.97
amarillo	1.00	0.88	1.00
oro	0.98	1.04	1.1
rosado	1.19	0.66	1.06
rojo	1.07	1.02	1.10
azul	1.07	0.82	1.09
morado	1.11	1.01	1.1
plata	1.1	1.15	1.08
verde	1.04	1.03	1.04
café	1.05	1.12	0.98
gris	1.1	1.25	1.07
negro	1.12	1.47	0.92

Además del color, el peso del vehículo, su tipo de carrocería y el uso que se le da también son factores determinantes en su comportamiento en la vía, y es diferente el impacto que tienen en un accidente. La Figura 9 muestra los tipos de vehículos contemplados en el Código de Seguridad Vial de Cuba [9].



FIGURA 9: Tipos de vehículos

Por lo general, los vehículos más grandes transitan a una menor velocidad por la vía, pues tienen mayor probabilidad de volcarse y de sufrir accidentes. La Tabla 2 contiene las velocidades máximas establecidas en Cuba (en kilómetros por hora) para cada tipo de vehículo en las diferentes vías.

Queda claro que un accidente donde esté involucrado un remolque será mucho más peligroso que un accidente entre dos vehículos de carga de pasajeros. Los conductores deben ser capaces de asumir la responsabilidad que lleva conducir un vehículo de tales dimensiones.

TABLA 2: Velocidades máximas según tipo de vehículo

Vehículo	Carretera	Autopista
Motor	50	
Automóviles de hasta 3500 kg y Ómnibus	90	100
Transporte de carga rígido y articulados	80	90
Remolques, grúas y similares	70	80
Construcción, industriales, tractores y equipos agrícolas	20	

Otra cuestión a tener en cuenta es el estado técnico que presenten los mismos. Aspectos mecánicos como el fallo en los frenos, en la dirección, en la suspensión, falta de mantenimiento, exceso de carga, modificaciones inadecuadas o sobredimensionamiento del vehículo pueden ser motivo de desperfectos sobre la marcha. La responsabilidad de los conductores y dueños de vehículos en hacer revisiones periódicas a los mismos, ayuda a una correcta circulación y evita accidentes.

E. Vehículos que adelantan carril

Una de las conductas más adoptadas por los conductores es la de adelantar a los vehículos que tienen por delante. Esta conducta puede estar condicionada por los factores anteriormente mencionados, así como por el simple hecho del conductor que querer llegar pronto a su destino. El conductor de un vehículo para adelantar a otro, en vías de dos o tres carriles, o sendas en ambos sentidos de circulación, está obligado a [9]:

- comprobar que puede realizar la maniobra sin interferencia a los demás vehículos que marchen delante, detrás o se acerquen en sentido opuesto al suyo, y sin riesgo de accidente;
- efectuar el adelantamiento por la senda izquierda;
- incorporarse de forma gradual y segura a la senda o carril por la que transitaba, siempre que no obligue al conductor del vehículo adelantado a modificar su dirección o velocidad.

El conductor del vehículo adelantado está obligado a [9]:

- no aumentar la velocidad, ni efectuar maniobras que impidan o dificulten el adelantamiento;
- facilitarle espacio suficiente al vehículo que lo adelanta, para reintegrarse a la senda o carril por la que transita.

F. Factor humano en la vía

Los errores humanos son una de las causas más frecuentes en los accidentes de tránsito y muchas veces pueden ser evitables. El exceso de confianza, la distracción, el consumo de drogas y alcohol, el exceso de velocidad, conducir con cansancio o fatiga, usar dispositivos electrónicos, de navegación o de radio, ejecutar maniobras peligrosas o adelantamientos indebidos son algunos motivos provocadores de accidentes que tienen como base al factor humano.

Los peatones como usuarios de la vía también cargan en ocasiones con la culpa de los accidentes, ya sea por cruzar intersecciones señalizadas sin obedecer, estar distraídos, hacer carreras delante

de los vehículos, entre otras acciones que ponen en peligro su vida. Además, el estado de ánimo también influye sobre este comportamiento.

De manera general, en las personas juega un papel muy importante la experticia que tengan en la vía, ya sea de conductor o de peatón. Conocer las leyes del tránsito y las intersecciones por las que se transita siempre ayuda a desenvolverse mejor.

1.1.3. Importancia de una correcta sincronización

Se puede concluir que, en aras de lograr una pequeña contribución a la disminución de indicadores como la contaminación provocada por la generación excesiva de gases de los vehículos, así como los accidentes en vías reguladas por semáforos o cualquier otra señalización vial, es importante conocer qué configuraciones aplicar a los mismos y dónde colocar las señales verticales, horizontales o los agentes de circulación, y con ello minimizar los embotellamientos y contribuir al ahorro de recursos materiales y humanos y gastos económicos innecesarios para el país.

1.2. Sistemas inteligentes y Simulación

1.2.1. Los agentes

Según Wooldridge [4], no existe una definición aceptada de forma universal del término “agente”, y de hecho existe debate y controversia sobre este tema por la variedad de criterios. Una definición tomada de Wooldridge es: “Un agente es un sistema computacional situado en un entorno dado, que es capaz de actuar de forma autónoma para cumplir sus objetivos” [4]. Se muestra en la Figura 10 un resumen del comportamiento de un agente inteligente.

También, la Fundación para los Agentes Físicos Inteligentes (FIPA por sus siglas en inglés) [19] define al agente como una entidad de software con un grupo de propiedades, entre las que se destacan: ser capaz de actuar en un ambiente, comunicarse directamente con otros agentes, estar condicionado por un conjunto de tendencias u objetivos, manejar recursos propios, ser capaz de percibir su ambiente y tomar de él una representación parcial, ser una entidad que posee habilidades y ofrece servicios, que puede reproducirse, entre otras [20].

El paradigma de agentes se puede considerar como una extensión intrínseca de la orientación a objetos, mediante la cual un agente representa un objeto que tiene control sobre su ejecución [21].

De forma general, varios autores reconocen en los agentes diversas propiedades, entre las que se destacan el ser autónomos, reactivos, proactivos y tener habilidad social [20].

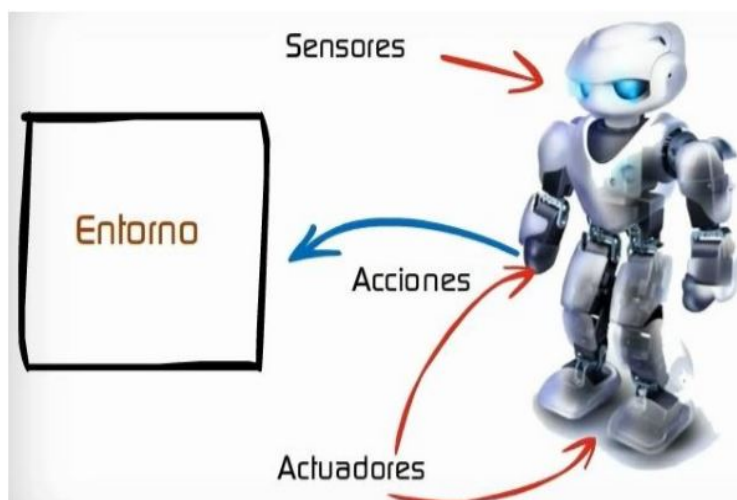


FIGURA 10: Comportamiento de un agente inteligente

1.2.2. Sistemas Multi-Agente

Los Sistemas Multi-Agente (MAS, *Multi-Agent Systems*), son conjuntos de agentes que interactúan entre sí, mediante el intercambio de mensajes. Los agentes poseen diferentes metas y motivaciones. En orden de interactuar satisfactoriamente y lograr sus objetivos, requieren las habilidades de cooperar, coordinar y negociar entre ellos [4].

1.2.3. Modelado y simulación

Varios filósofos (por ejemplo, Hesse (1963) y Hughes (1997)) que han estudiado la metodología científica tradicional han propuesto el mismo esquema general del proceso de modelado. Según estos autores, los modelos científicos se construyen para desarrollar procesos de inferencia sobre ciertos aspectos de sistemas reales previamente observados. Es mediante estos procesos de inferencia, mediante la construcción y el uso de modelos científicos, como mejoramos nuestro entendimiento de los sistemas reales observados. Es decir, una eficaz forma de mejorar nuestro conocimiento del mundo en que vivimos es a través de la creación de modelos, no necesariamente formales. El proceso de creación y uso de un modelo se muestra esquematizado en la Figura 11, la cual muestra un proceso secuencial con claridad, pero, hay que tener en cuenta que el proceso de modelado contiene en general varios bucles de retroalimentación.

La simulación se puede definir como la creación de un modelo computacional de un sistema real o propuesto, sobre el cual se realizan experimentos que permitan describir el comportamiento observado o futuro [2]. Es esencialmente una técnica de muestreo estadístico controlado, que puede usarse para estudiar el desempeño de sistemas informáticos. En una simulación se utiliza

una computadora para evaluar un modelo numéricamente y recolectar datos para estimar las características del modelo [1].

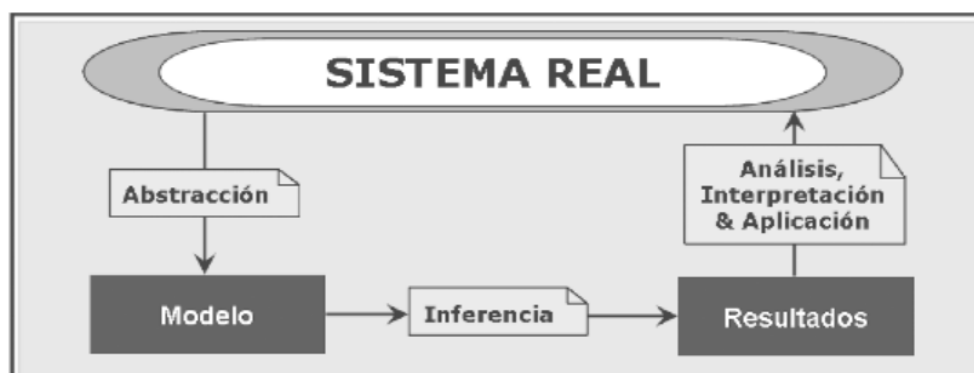


FIGURA 11: Esquema general del proceso de modelado científico

La simulación no resuelve los problemas por sí misma, sino que ayuda a identificar los problemas relevantes y a evaluar cuantitativamente las soluciones alternativas [2].

1.2.4. Simulación Basada en Multi-Agente

Como se había mencionado anteriormente, un agente se ajusta perfectamente a las demandas de aplicaciones complejas e inherentemente distribuidas, donde cada agente es un sistema computacional en sí mismo, así como también modificable independientemente [22].

La Simulación Basada en Agentes (ABS) se refiere a una categoría de modelos computacionales que invoca las acciones dinámicas, las reacciones y los protocolos de intercomunicación entre los agentes en un entorno compartido, para evaluar su diseño y rendimiento y obtener información sobre su comportamiento y propiedades emergentes. Desde el punto de vista de la simulación, la función de un componente individual puede variar desde reglas reactivas muy básicas hasta modelos de comportamiento cognitivos más sofisticados.

El objetivo de la ABS es modelar sistemas complejos que adoptan un enfoque ascendente a partir de los agentes individuales [23]. Un enfoque concreto de la ABS es modelar y simular escenarios realistas con un grupo de agentes autogobernados, ya sea como entidades simples dentro de los fragmentos de códigos de cómputo o como objetos considerablemente inteligentes. Esto posiblemente se considere sinónimo de capacidades de resolución de problemas del ser humano con estados infinitos, creencias, confianzas, decisiones, acciones y respuestas. Adquirir el conocimiento adecuado del sistema para construir un modelo conceptual y lógico apropiado es una de las tareas más desafiantes de la simulación [24].

1.2.5. Modelos matemáticos

Un modelo matemático es un modelo que utiliza fórmulas matemáticas para representar la relación entre distintas variables, parámetros y restricciones. Los modelos matemáticos son utilizados para analizar la relación entre dos o más variables. Pueden ser utilizados para entender fenómenos naturales, sociales, físicos, etc. Dependiendo del objetivo buscado y del diseño del mismo modelo pueden servir para predecir el valor de las variables en el futuro, hacer hipótesis, evaluar los efectos de una determinada política o actividad, entre otros objetivos [25].

Propiedades deseadas de un modelo matemático

Cuando se diseña un modelo matemático, se busca que este tenga un conjunto de propiedades que ayude a asegurar su robustez y efectividad. Entre estas propiedades se encuentran [25]:

- **Simplicidad:** Uno de los objetivos principales de un modelo matemático es simplificar la realidad para poder entenderla mejor.
- **Objetividad:** Que no tenga sesgos ni teóricos ni de los prejuicios o ideas de sus diseñadores.
- **Sensibilidad:** Que sea capaz de reflejar los efectos de pequeñas variaciones.
- **Estabilidad:** Que el modelo matemático no se altere significativamente cuando hay cambios pequeños en las variables.
- **Universalidad:** Que el modelo sea aplicable para varios contextos y no sólo a un contexto en particular.

Clasificaciones de modelos matemáticos

Según tipo de representación

- **Cualitativos o conceptuales:** Se refieren a un análisis de la calidad o la tendencia de un fenómeno sin calcular un valor exacto.
- **Cuantitativos o numérico:** Los resultados obtenidos tienen un valor concreto que tiene un cierto significado (puede ser exacto o relativo).

Según la aleatoriedad

- **Determinista:** No tiene incertidumbre, los valores son conocidos.
- **Estocástico:** No se conoce con exactitud el valor de las variables en todo momento. Existe incertidumbre y por ende una distribución de probabilidad de los resultados.

Según su aplicación u objetivo

- Simulación o descriptivo: Simula o describe un fenómeno. Los resultados se enfocan a predecir qué sucederá una determinada situación.
- Optimización: Se utilizan para encontrar una solución óptima a un problema.
- De control: Para mantener el control de una organización o sistema y determinar las variables que deben ajustarse para obtener los resultados buscados.

El modelo matemático puede decir entonces que el problema es elegir los valores de las variables de decisión de manera que se maximice la función objetivo, sujeta a las restricciones dadas. La determinación de los valores apropiados que deben asignarse a los parámetros del modelo, un valor por parámetro, es una tarea crítica y a la vez un reto en el proceso de construcción del modelo [1].

1.2.6. Modelos basados en lógica difusa

La incertidumbre en algunos casos se debe a ausencia de información o desconfianza en la fuente de la misma, pudiendo en algunos casos asociarse esta a un valor subjetivo y en otros casos dándole un significado probabilístico (asociado a la probabilidad de ocurrencia del fenómeno analizado). Sin embargo, en determinadas ocasiones, la incertidumbre se presenta en casos en que la información es totalmente confiable, pero el problema está en que no se sabe bien cómo clasificar la información recibida. Por ejemplo, es muy difícil asignar valores de verdad absolutos para clasificar a los individuos en Altos, Delgados o Alegres [26]. La teoría de los conjuntos difusos es un instrumento útil en estos casos, para la especificación de lo bien que un objeto satisface una descripción vaga [27].

Por ejemplo, considere la proposición «Pedro es experto condiciendo». ¿Es esto verdad si Pedro conduce hace 5 años? La mayoría de la gente vacilaría al responder «verdadero» o «falso», preferirían decir, «más o menos». Nótese que esto no es una pregunta de incertidumbre sobre el mundo externo (estamos seguros de los años de experiencia de Pedro). El asunto es que el término lingüístico «experto» no se refiere a una delimitación clara de los objetos en dos clases (hay grados de experticia).

Por esta razón, la teoría de los conjuntos difusos no es un método para razonamiento incierto en absoluto. Mejor dicho, la teoría de los conjuntos difusos trata los clasificadores como predicados difusos y dice que el valor de verdad es un número entre 0 y 1, en vez de un verdadero o falso exacto. El nombre «conjunto difuso» proviene de la interpretación del predicado como la definición implícita de un conjunto de sus miembros (un conjunto que no tiene fronteras delimitadas) [27].

La lógica difusa es un método para el razonamiento con expresiones lógicas que describen las pertenencias a los conjuntos difusos. La lógica difusa es así un sistema de operatividad de la verdad, un hecho que causa serias dificultades [27]. Es multivaluada y permite representar matemáticamente la incertidumbre y la vaguedad, proporcionando herramientas formales para su tratamiento [28].

En general la lógica difusa se aplica a sistemas de control y para modelar cualquier sistema continuo de ingeniería, física, biología o economía. Las aplicaciones van desde procesos tan complejos como el control de un reactor, el movimiento de una grúa, o la modelación de la decisión de un experto (Sistemas Expertos Difusos), hasta cosas tan cercanas a todos como son las lavadoras que tienen control automático del agua, el detergente y el desagüe, y que pueden identificarse claramente porque dicen "*Fuzzy Logic*". Este es quizás una de las aplicaciones más conocidas y populares de las técnicas de Inteligencia Artificial [27].

1.3. Sistemas de Información Geográfica

Los Sistemas de Información Geográficos (GIS, *Geographic Information System*) constituyen sistemas integrados y organizados de hardware, software y datos geográficos, diseñados para capturar, almacenar, manipular, analizar y desplegar información geográficamente referenciada y además representar la misma, con el fin de resolver problemas complejos de planificación y gestión [29].

Están compuestos por una serie de elementos, donde los datos espaciales son esenciales debido a que contienen la información geográfica necesaria para el funcionamiento de estos sistemas. Esta información puede ser almacenada tanto en ficheros como en bases de datos espaciales, utilizándose para dicho fin varias tecnologías, y las más comunes son las bases de datos relacionales [30].

Un ejemplo de almacenamiento en bases de datos puede ser mediante SQLite [31]. Para esto es necesario utilizar la extensión espacial SpatiaLite [32], una biblioteca de código abierto que brinda soporte completo a capacidades espaciales de tipo SQL. Permite el almacenamiento de diferentes geometrías y su consulta a través de un conjunto de métodos almacenados. Igualmente, posee procedimientos para efectuar una gran cantidad de operaciones espaciales, como pueden ser el cálculo de distancia, área, centroide y rutas, por solo citar ejemplos [32].

Existen dos modelos para representar los datos espaciales, ráster y vectorial, siendo común que ambos coexistan en un mismo sistema [30]. El modelo ráster consiste en una matriz rectangular de celdas cuadradas o píxeles usados para representar información relativa a capas temáticas, elevaciones e imágenes satelitales, entre otras [33, 30]. A su vez, el modelo vectorial se basa en el uso de puntos independientes con atributos matemáticos asociados [34, 33], como unidad primitiva, que se agrupan para formar geometrías más complejas. Es utilizado para la

representación de aspectos pocos variables y cualitativos, como la representación geométrica de territorios y la altitud [35]. En la actualidad, existen varias entidades que proveen datos espaciales. Una de estas es Open Street Map (OSM) [36].

Los datos primitivos o elementos básicos de la cartografía de OSM son:

- Los nodos (*nodes*). Son puntos que recogen una posición geográfica dada.
- Las vías (*ways*). Son una lista ordenada de nodos que representa una polilínea o un polígono (cuando una polilínea empieza y finaliza en el mismo punto).
- Las relaciones (*relations*). Son grupos de nodos, vías u otras relaciones a las que se pueden asignar determinadas propiedades comunes. Por ejemplo, todas aquellas vías que forman parte del Camino de Santiago.
- Las etiquetas (*tags*). Se pueden asignar a nodos, caminos o relaciones y constan de una clave (*key*) y de un valor (*value*). Por ejemplo: *highway=trunk* define una vía como carretera troncal.

La cobertura de datos se extiende al conjunto de todos los países del mundo, siendo esta cada vez mayor en las regiones emergentes del planeta. El crecimiento es sorprendentemente constante, lo que significa una gran cantidad de personas trabajando activamente en aportar datos al mapa.

1.4. Herramientas de configuración de señales existentes en el mundo

Existen varias herramientas de configuración de semáforos que se basan en la interconexión de los mismos mediante una red, como es el caso de [37] y [38] que diseñan un sistema unificado de conmutación de semáforos a través de una intranet con interconexión hacia todos los sitios remotos en el ejercicio de mejorar el tráfico vehicular en la ciudad.

En las últimas décadas, muchas agencias e investigadores planificadores del transporte han intentado mejorar los sistemas de señales desplegados tratando de poner a punto los parámetros de control. Uno de los métodos mayormente usados en aplicaciones ingenieriles es aplicar un acercamiento de optimización basado en simulación con respecto a las estrategias y políticas pre-determinadas [39].

Por ejemplo, en el año 2007 se presenta una simulación para intersecciones señalizadas bajo un ambiente tridimensional y pretende obtener como resultado de ello los tiempos de espera a los que se ven sujetos los conductores en dichas intersecciones, así como también observar visualmente el comportamiento de los vehículos de forma individual y como conjunto, el sistema

en sus partes y como un todo. En este trabajo se usa una intersección genérica construida por la herramienta de modelado 3D utilizada [40].

Una visión más matemática se documenta en [41], donde se diseñó e implementó un simulador de flujo vehicular que incorpora un método para realizar la sincronización de un circuito de calles. El método de sincronización desarrollado en este trabajo se realiza por medio de varias teorías como son: teoría de colas, ecuaciones diferenciales parciales, teorías de ondas, las ondas de Shock; todas estas teorías conjuntadas proporcionan el tiempo de duración que debe de tener cada ciclo de los semáforos que conforman el circuito de calles. Por medio del concepto de offset se calculan los tiempos que deben transcurrir entre el inicio del ciclo verde de un semáforo y del semáforo que le sigue.

El simulador THE [42], implementado en Pakistán ofrece una solución al problema del tiempo excesivo de espera de los vehículos en intersecciones señalizadas. Utiliza un algoritmo genético para optimizar el tiempo de luz verde de cada semáforo. Este simulador tiene varias limitaciones; trabaja con un número fijo de vehículos a simular, no utiliza mapas reales y solo acepta movimientos en línea recta de los vehículos. Pero en principio fue un paso de avance para entrar en el campo de las simulaciones de tráfico.

El software Arena también se usó para simular los procesos del tráfico, modelando vehículos que se comporten según las indicaciones de los semáforos, teniendo en cuenta no solo las posibilidades de pasar o detenerse, sino también las opciones de seguir en línea recta o doblar hacia alguna otra dirección. Con ello logran mostrar que este software también puede ser usado para simulaciones de tránsito [43].

A partir de [44] comienzan a aparecer en las simulaciones términos de inteligencia artificial como los agentes. El artículo presenta el desarrollo de una metodología novedosa que permite incluir un modelo formal basado en agentes autónomos e inteligentes capaces de manipular las fases de los ciclos en una infraestructura de semáforos de acuerdo a las exigencias y limitaciones de la carretera. Este proceso es mejora efectiva e inmediata de la calidad del servicio en una intersección, aumentando el rendimiento de la movilidad de los vehículos y mejorando la generación de emisiones, cuando los vehículos se paran en un semáforo rojo. La mayor limitación de esta herramienta es no utilizar mapas reales.

Las metaheurísticas también se han usado con el objetivo de optimizar fenómenos de tráfico. Tal es el caso de [45, 46] que presentan dos versiones de una herramienta para establecer una política de cambio en los tiempos de las luces de los semáforos en horas picos, usando modelos probabilísticos utilizando algoritmos metaheurísticos de tipo Recocido Simulado, el problema de optimización, mono-objetivo o multi-objetivo resultante de minimizar una o varias medidas de congestión. Mediante una simulación se evalúa la solución propuesta.

El fenómeno del tráfico es estudiado por otras ramas ingenieriles como muestra [47], donde el Departamento de Ingeniería Mecánica de la Universidad Politécnica de Madrid lo usa para

simular de forma eficaz el movimiento de una serie de vehículos y las interacciones entre ellos y otros elementos de trayecto y así predecir en qué momento se puede usar cada tipo de conducción y el consumo de combustible necesario en ellas.

Los simuladores sobre mapas e intersecciones genéricas fueron un buen comienzo en los estudios sobre el tráfico, pero se hacía necesario utilizar mapas obtenidos a partir de una base de datos espacial real. La herramienta desarrollada por [48] utiliza OSM para sus simulaciones. La idea principal es prever hacia que calles se desvía el tráfico después de configurar el semáforo de un cruce. Esta herramienta es muy costosa computacionalmente debido a que incluye planificación automática.

También se usa OSM en [49], herramienta que permite editar la información de cualquier mapa, incluyendo las calles, detectores, etc. con el objetivo de mejorar el comportamiento de vehículos y semáforos.

Otra de las ramas de la inteligencia artificial es la Lógica Difusa, utilizada en [39] para aproximar el razonamiento humano mientras exista incertidumbre para modelar las condiciones del tráfico percibidas. Esta herramienta incorpora además conocimientos automáticos y electrónicos para en tiempo real conectar con los semáforos y alterar sus parámetros iniciales.

Otro simulador que centra su atención en fenómenos viales es SUMO [50]. SUMO es un paquete de simulación de tráfico de código abierto diseñado para simular cómo se desarrolla una demanda de tráfico dada, que consiste en movimientos de vehículos a través de una red de carreteras predeterminada. SUMO es microscópico, en donde cada vehículo es modelado explícitamente, tiene un camino propio y se mueve individualmente a través de la red. Uno de los mejores módulos de SUMO es el protocolo de Interfaz de Control de Tráfico (TraCI, *Traffic Control Interface*) [51], que es una API que trata la simulación de SUMO como un servidor y permite a los usuarios obtener información de la simulación de tráfico o modificar la simulación. Durante la simulación, los nuevos vehículos son generados por la distribución uniforme encima de 20 secciones de la entrada. En esta simulación, el número de fases es 4, uno para permitir los vehículos circulantes a cada una de las cuatro direcciones.

De manera general las herramientas desarrolladas hasta este momento en el mundo solo tienen en cuenta los vehículos y semáforos como componentes de la simulación, sin contemplar que el comportamiento del tráfico depende de un gran número de factores externos como el estado del pavimento o la interrupción de la vía por obstáculos. Si bien es cierto que en los países desarrollados estos factores ocurren en menor medida, para tener una simulación lo más cercana a la realidad posible es necesario incorporarlos.

1.5. Conclusiones parciales

Al concluir el presente capítulo se arriban a las siguientes conclusiones:

- Las malas configuraciones de las señales en la vía traen como consecuencia embotellamientos, accidentes y contaminación ambiental.
- El comportamiento de los usuarios de la vía está condicionado por factores externos como son el estado del pavimento, las características y el estado técnico de los vehículos en los que transitan, la temperatura ambiental, los peatones y animales que interrumpen la vía y la experticia de los usuarios del tráfico.
- La Simulación Basada en Multi-Agente es un método eficaz para describir sistemas complejos como el transporte, y analizar y comprender el comportamiento de los mismos.
- La lógica difusa ayudará a escalar los niveles de experticia de los conductores y los peatones.
- Para la captura, almacenamiento, manipulación y análisis de información geográfica se utilizan los GIS.
- Las herramientas desarrolladas en el mundo no contemplan todos los factores influyentes en el fenómeno del tráfico como la sensación ambiental, el pavimento o las características de los vehículos.
- La confección de un modelo matemático que apoye a la simulación es de gran importancia para la comprensión de la misma.
- Es necesario un mecanismo de evaluación más real para las configuraciones semafóricas que permita conocer el funcionamiento que tendrán las señalizaciones viales una vez colocadas.

CAPÍTULO 2

Herramienta de Simulación de Tráfico

En el presente capítulo se documenta toda la propuesta de solución. Se exponen los elementos que forman parte del dominio, la relación entre las entidades y las reglas del negocio. Se presentan los diagramas en UML (UML, *Unified Modeling Language*) [52], necesarios para explicar el flujo de las actividades que forman parte de la arquitectura de la herramienta de simulación, así como los casos de uso, diagramas de clases y modelo físico de la base de datos. Se muestran además los requisitos identificados para el software y las políticas seguidas para la interfaz de usuario, los patrones utilizados durante de desarrollo de la herramienta.

2.1. Modelado del negocio

Los modelos se crean con el objetivo de entender mejor la entidad real que se va a construir. Un modelo debe ser capaz de representar la información que el software transforma, la arquitectura y las funciones que permiten que esto ocurra, las características que desean los usuarios y el comportamiento del sistema mientras la transformación tiene lugar. Los modelos deben cumplir estos objetivos en diferentes niveles de abstracción, en primer lugar, con la ilustración del software desde el punto de vista del cliente y después con su representación en un nivel más técnico [53].

2.1.1. Modelo del dominio

Debido a que la solución no modela procesos de negocio establecidos, se posee conocimiento de la problemática y el resultado de llevar a cabo la implementación de la propuesta es un componente, se hace necesaria su representación con un modelo de dominio.

El modelo de dominio es una representación de las clases conceptuales del mundo real, no de componentes de software. Se muestran los conceptos significativos en un dominio del problema [54]. El diagrama de las entidades del dominio se presenta en la Figura 12.

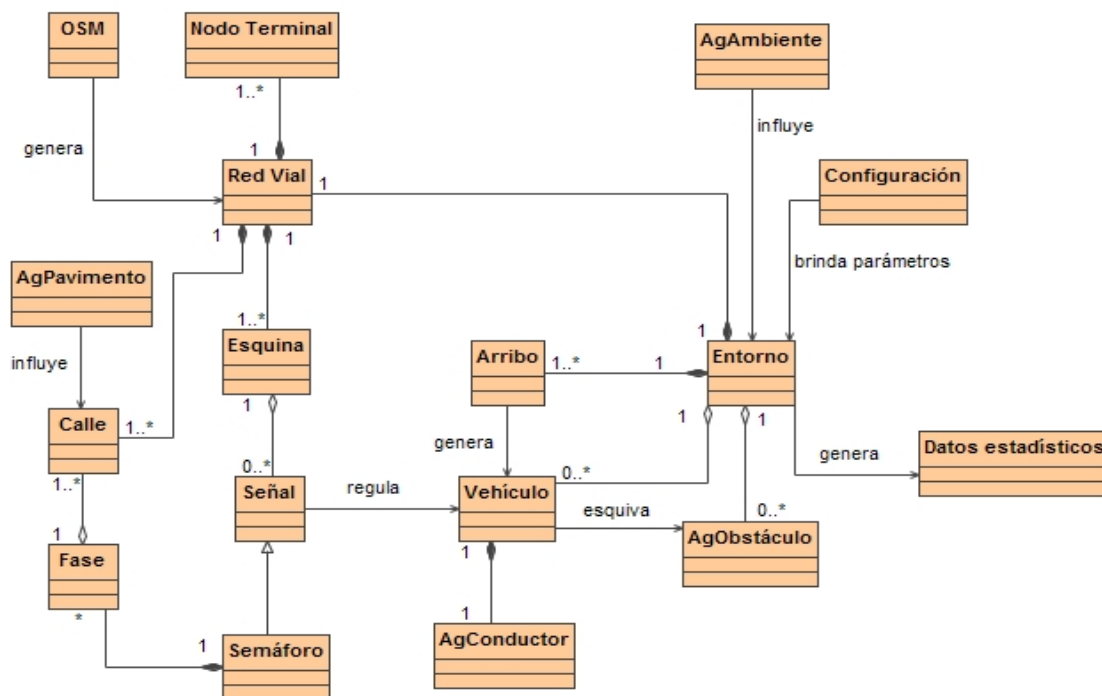


FIGURA 12: Modelo de dominio

Definición de clases y conceptos

En la Tabla 3 y la Tabla 4 se muestra el significado de cada una de las entidades que forman parte del modelo de dominio de la solución propuesta y la relación entre ellas.

TABLA 3: Definición de los conceptos del Modelo del Dominio (1)

Clases	Descripción
OSM	Fichero que contiene la información espacial, a partir de esta se obtiene la información necesaria para crear las redes.
Red Vial	Son los datos espaciales de las calles y las esquinas. Es generada a partir del OSM.
Nodo Terminal	Punto donde empieza o finaliza solo una calle.
Calle	Vía por la que se trasladan los vehículos. Estas entidades son uno de los elementos principales de una red vial.
Pavimento	Agente que controla el estado del pavimento de las calles.
Esquina	Punto de intersección entre dos o más calles. Lugar donde son colocadas las señales de control del flujo del tráfico
Señal	Representa tanto a las señales verticales como a las horizontales. También puede representar a los oficiales del tránsito o a cualquier otro controlador de tráfico
Semáforo	Constituye una señal de tránsito. Se sitúa en una esquina y regula el paso de vehículos por esta.

TABLA 4: Definición de los conceptos del Modelo del Dominio (2)

Clases	Descripción
Fase	Varias fases componen un semáforo. Estas tienen un estado que indica que luz está habilitada. A cada fase se le hace corresponder una o varias calles.
Conductor	Representa a la persona que maneja un vehículo con datos como la edad, el sexo y la cantidad de años de experiencia manejando.
Vehículo	Entidades autónomas. Se desplazan por una red vial tomando decisiones como las rutas, la velocidad, el carril de la calle, pueden respetar o no las señales, o incluso pueden romperse o colisionar entre ellos.
Arribo	Se corresponde a un nodo terminal, su función es generar vehículos dependiendo de un tiempo entre arribo previamente determinado. .
Entorno	Espacio donde interactúan los agentes.
Ambiente	Agente que modifica datos ambientales como la lluvia, la humedad, la hora del día y la temperatura.
Obstáculo	Representa los obstáculos que pueden existir en la vía como son animales, peatones u objetos.
Datos estadísticos	Datos almacenados al finalizar la simulación. Son el resultado de la misma.
Configuración	Parámetros de simulación pasados al entorno antes de comenzar la misma.

2.1.2. Reglas del Negocio

Existen un conjunto de restricciones para obtener un correcto funcionamiento de los negocios y sistemas ante la ocurrencia de eventos inesperados y no calculados. Estas restricciones se conocen como reglas del negocio [54]. A continuación, se exponen las reglas del negocio presentes en la solución.

- Para realizar una reconfiguración deben haberse almacenado los resultados estadísticos de la simulación anterior.
- El mapa tiene calles.
- Las señales controladoras del flujo del tráfico (“Pare”, “Ceda el Paso”, Semáforo) solo se pueden colocar en las esquinas.
- Solo cuando haya instalado un semáforo en una esquina se podrá poner una señal de “Ceda el Paso con luz roja” o “Ceda el Paso con luz verde”.
- Solo cuando haya instalado un “Reductor de Velocidad” se podrá poner una señal de proximidad del mismo.
- Solo antes de comenzar la simulación el estado del pavimento se podrá modificar en el escenario de la misma.

- Solo después de finalizada la simulación se pueden analizar los datos generados.
- Solo cuando se haya seleccionado un mapa este se visualizará.
- Es responsabilidad del analista configurar los parámetros de la simulación.
- Mientras no se esté satisfecho con los resultados de la simulación se puede reconfigurar y volver a simular ese escenario.
- Las fases de los semáforos solo pueden tomar los valores “rojo”, “amarillo” y “verde”.
- Existe probabilidad de que los vehículos cometan infracciones.
- Existe probabilidad de que los vehículos puedan colisionar.
- Existe probabilidad de que los vehículos adelanten a otros vehículos en su carril.
- Existe probabilidad de que los vehículos sufran desperfectos técnicos.

2.1.3. Diagrama de Actividades

En la Figura 13 es el Diagrama de Actividades que recoge una vista dinámica de alto nivel del flujo principal del proceso.

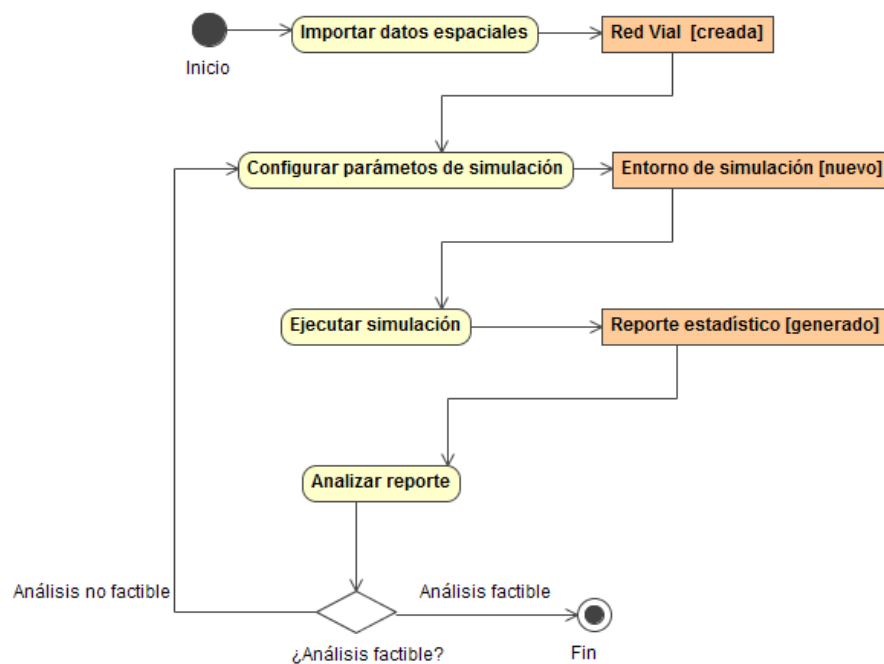


FIGURA 13: Diagrama de actividades

2.2. Captura de Requisitos

En la Tabla 5 se identifican los requisitos candidatos para el análisis de los requisitos funcionales. Se tienen en cuenta la prioridad, con un rango de 1-4 (donde 4 es la máxima) y también el tipo de requisito (normal, esperado, innovador) [7].

TABLA 5: Identificación de requisitos

Prioridad	Nombre del requerimiento	Tipo de requisito
4	Configuración	Normal
3	Pausar simulación	Innovador
2	Exportar datos estadísticos	Normal
2	Interfaz agradable	Esperado
1	Multiplataforma	Innovador
3	Eficiencia	Esperado
3	Análisis de datos	Esperado
1	Reporte	Esperado
3	Rendimiento	Esperado

2.2.1. Requisitos Funcionales

- Utilizar datos espaciales para modelar una red vial.
- Representar visualmente las vías, semáforos, señales verticales y horizontales, vehículos y otros elementos que forman parte de la simulación.
- Gestionar los atributos de los semáforos, el ambiente, los vehículos y arribos desde la interfaz de usuario.
- Configurar los parámetros de la simulación desde la interfaz de usuario.
- Representar visualmente el estado del semáforo utilizando los colores preestablecidos (verde, amarillo y rojo).
- Representar visualmente la existencia de las señales en el escenario de simulación.
- Agregar comportamiento a los vehículos (agentes) para que eviten colisionar.
- Agregar comportamiento a los vehículos (agentes) para que actúen según el estado del semáforo y demás señales.
- Agregar comportamiento a los vehículos (agentes) para que cometan infracciones.
- Agregar comportamiento a los vehículos (agentes) para que adelanten carril.

- Agregar comportamiento a los vehículos (agentes) para que eviten los obstáculos.
- Agregar comportamiento a los vehículos (agentes) para que puedan romperse durante la marcha.
- Agregar comportamiento a los vehículos (agentes) para que actúen según las características de su conductor.
- Agregar comportamiento a los vehículos (agentes) para que actúen según las características del ambiente.
- Agregar comportamiento a los vehículos (agentes) para que actúen según las características del pavimento.
- Registrar los datos estadísticos vinculados con la congestión del tráfico para apoyar el análisis y la toma de decisiones.

2.2.2. Requisitos No Funcionales

En la Tabla 6 y la Tabla 7 se muestran los Requisitos No Funcionales definidos para el software a desarrollar, que fueron identificados a partir de los problemas frecuentes también presentes en las tablas.

TABLA 6: Requisitos No Funcionales (1)

Requisito	Problemas frecuentes	Descripción
Rendimiento	<ul style="list-style-type: none"> ■ Alta disponibilidad ■ Implementación del acceso a los datos ■ Control de las transacciones ■ Registro de eventos diarios del sistema (<i>Logs</i>) 	El sistema debe funcionar de manera eficiente, que el sistema se ralentice por cuestiones de rendimiento podría significar una variación en los datos, por lo que para asegurarnos de generar datos correctos la aplicación debe tener un buen rendimiento.
Portabilidad	<ul style="list-style-type: none"> ■ Estructuración y subdivisión del sistema en partes pequeñas 	La herramienta debe admitir multiplataforma para que de esta forma pueda ser ejecutada desde diferentes Sistemas Operativo (OS, <i>Operative System</i>)

TABLA 7: Requisitos No Funcionales (2)

Requisito	Problemas frecuentes	Descripción
Apariencia y usabilidad	<ul style="list-style-type: none"> ■ Control de la navegación del usuario ■ Control y atención a los eventos generados por el usuario 	Tener una buena apariencia es clave en este tipo de herramienta ya que se comprende mejor el entorno que queremos modelar. En cada momento se debe brindar retroalimentación del estado de los procesos.
Tratamiento de excepciones	<ul style="list-style-type: none"> ■ Tratamiento de excepciones ■ Tolerancia a fallos 	Cualquier excepción que se lance en la ejecución de la herramienta debe ser tratada acorde a una política que impida pérdida de datos y notificada al usuario para que pueda buscar en ayuda y soporte técnico.
Requisitos de software		Se requiere disponer en la estación de trabajo sobre la cual se va a ejecutar el sistema la Máquina Virtual de Java versión 6.0 o superior.

2.3. Evaluación de arquitecturas candidatas

Para comenzar el desarrollo de una herramienta se hace necesario seleccionar con qué arquitectura trabajar. Ello implica realizar de estudios de todas las arquitecturas que puedan servirle a la solución que se propone y analizar sus ventajas y desventajas.

Se deben evaluar los atributos de calidad considerando cada atributo de forma aislada y analizar las arquitecturas candidatas utilizando el análisis de sensibilidad [53].

2.3.1. Plataformas de agentes

En el Anexo A se realiza un estudio de algunas de las plataformas de agentes actuales escogidas entre las más reconocidas según FIPA [19] para el desarrollo de herramientas de simulación que cuenten con la presencia de agentes inteligentes.

Luego de hacer una síntesis de los detalles más importantes de algunas plataformas de agentes disponibles, la Tabla 8, muestra una comparación resumen teniendo en cuenta las características necesarias que más se adecuan a la problemática [24].

TABLA 8: Comparación de las plataformas de agentes

Plataforma	Lenguaje	GIS	Licencia	Soportes para usuario
MASON	Java	Si	Académica libre	API, tutoriales, documentación, listas de correo y ejemplos
JADE	Java	No	LGPL	Documentación, tutoriales y ejemplos
JADEx	Java	No	GNU- LGPL	Documentación, tutoriales y ejemplos
NetLogo	NetLogo	Si	GPL	Documentación, FAQ, tutoriales y listas de correo
RePast	Java, C++, C#	Si	BSD	Lista de correo, tutoriales, FAQ y ejemplos
GAMA	GAML	Si	BSD	Videos tutoriales y ejemplos

2.3.2. MASON como plataforma de simulación

La herramienta precedente a la desarrollada en este trabajo utilizó la arquitectura MASON, plataforma de simulación basada en agentes que brinda facilidades para el trabajo con GIS y la manipulación de eventos discretos. No requiere de mucha capacidad de cómputo para ejecutar las simulaciones y según una comparación entre plataformas realizado en el año 2017 [24], es una de las plataformas más usadas para simular problemas de transporte. A continuación, se detallan sus características.

Estilos y patrones

Un estilo arquitectónico es una transformación que se impone al diseño de todo el sistema. El objetivo es establecer una estructura para todos los componentes del sistema. Se utilizan con patrones arquitectónicos para el software que definen un enfoque específico para el manejo de algunas características del sistema [53]. Los utilizados por MASON son los siguientes:

Estilo: Centrado en Datos. **Patrón:** Repositorio

La Figura 14 ilustra los módulos de la solución que acceden a la base de datos centralizada con que cuenta la herramienta. La información para representar la red vial es obtenida de la base de datos espacial y luego los resultados de la simulación son almacenados.

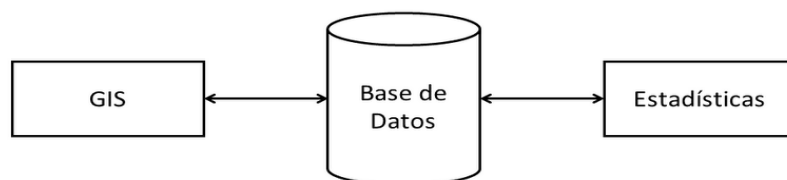


FIGURA 14: Arquitectura MASON, Patrón Repositorio

Estilo: Llamada y retorno. **Patrón:** Modelo Vista Controlador (MVC, *Model-View-Controller*).

En la Figura 15, se ve una evidente separación de los modelos, los controladores y las vistas y muestra además como ocurren las interacciones entre ellos [55].

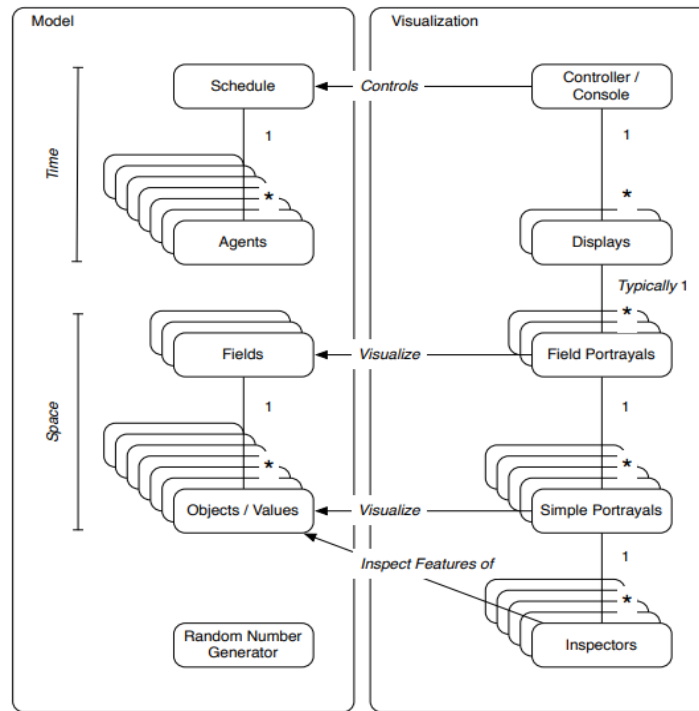


FIGURA 15: Arquitectura MASON, Patrón MVC

Estilo: Flujo de Datos. **Patrón:** Filtros y tuberías

La Figura 16 muestra el flujo de los datos desde que ingresan al sistema.

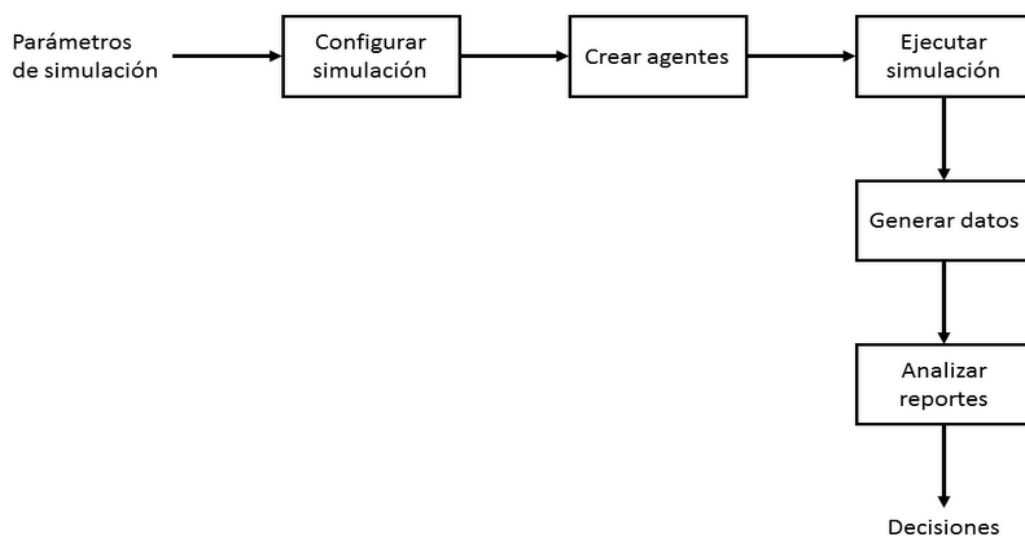


FIGURA 16: Arquitectura MASON, Patrón Flujo de Datos

Vista general de la arquitectura

MASON se divide en dos partes. La primera parte es el modelo, y la segunda parte es la visualización, actualmente en 2D o en 3D. Excepto cuando se elige tener objetos de modelo que se muestren, el modelo y la visualización están totalmente separados. Esto significa que el modelo se puede ejecutar sin visualización; ejecutar con visualización de diferentes tipos; y tener su visualización cambiada, añadida o eliminada en cualquier momento. El modelo también puede ser *checkpointed*, lo que significa que puede ser congelado y escrito en el disco, para descongelarse y continuar incluso en otro tipo de equipo [55]. En la Figura 17 se muestra una visión general de la arquitectura MASON.

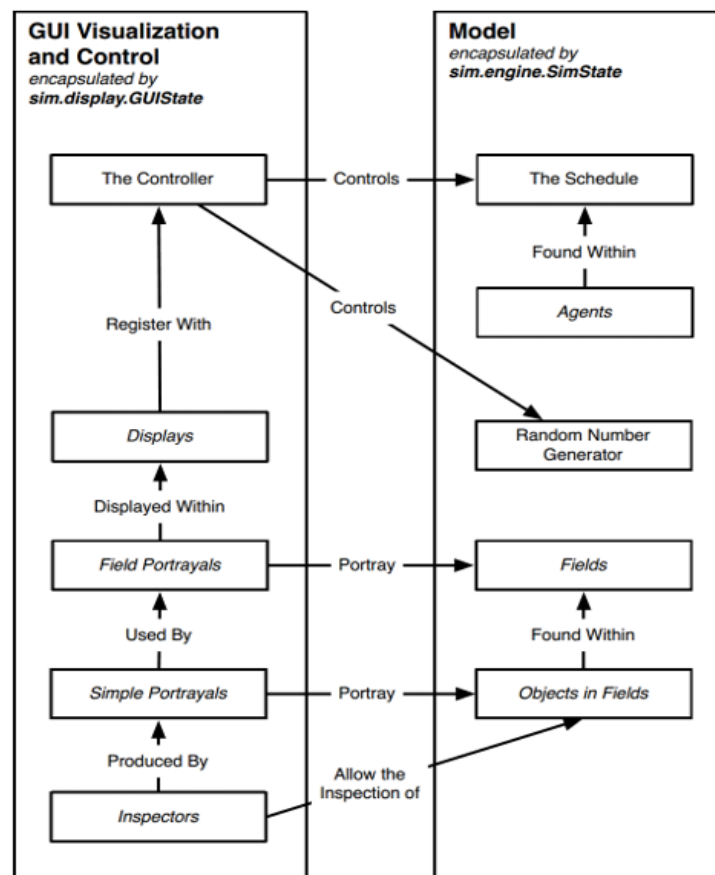


FIGURA 17: Relaciones primarias entre el modelo y la GUI visualización / control

Vista detallada de la arquitectura

El modelo de simulación está totalmente encapsulado dentro de una sola clase: una subclase de *sim.engine.SimState*. MASON creará una única instancia de su subclase para mantener el modelo. El propósito de esta clase es darle un lugar para almacenar todos y cada uno de los elementos que considere necesarios para su simulación.

MASON puede serializar todo el modelo a un archivo de punto de control, lo que significa que puede congelar la simulación en el tiempo y guardarlo en su totalidad en el archivo. Puede reiniciar el modelo desde este archivo, incluso en un equipo diferente o en un sistema operativo, o bajo una facilidad de visualización, y la simulación continuará donde lo dejó sin variación alguna. Para que esto sea posible, todos los objetos del modelo MASON son `java.io.Serializable`, lo que significa que pueden ser (en gran parte) automáticamente escritos o leídos de un flujo. Esto está descrito en la Figura 18 [55].

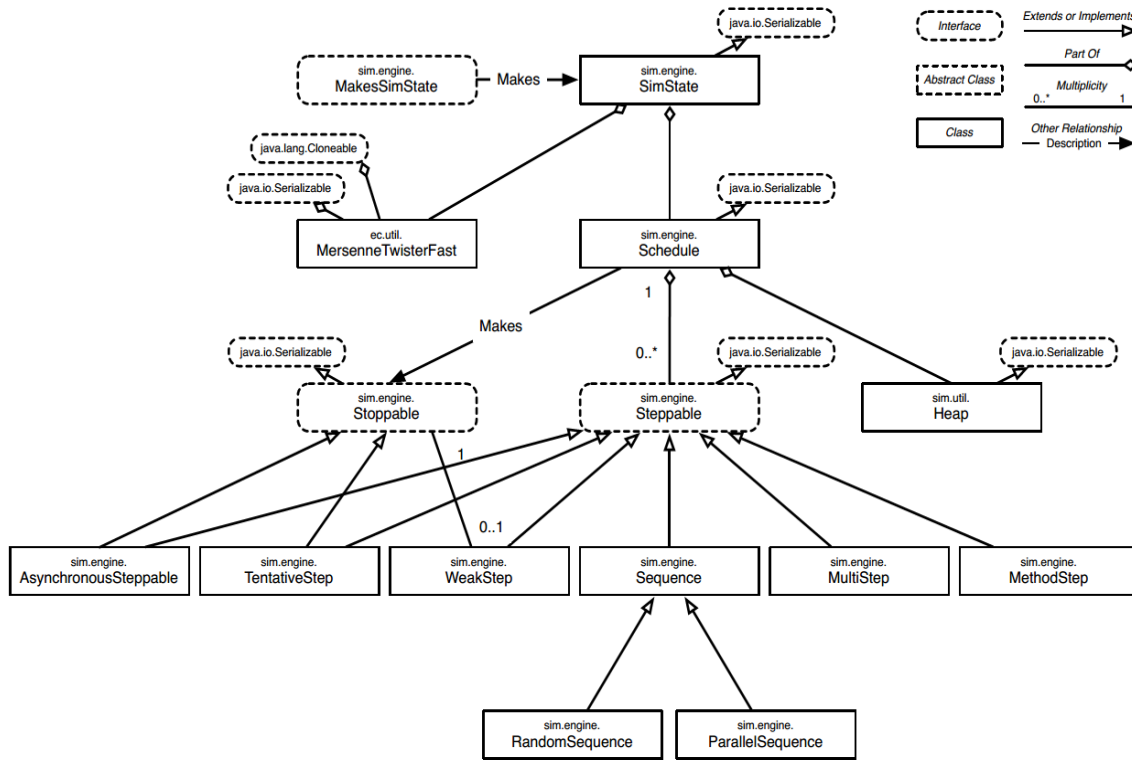


FIGURA 18: Diagrama de la simulación de núcleo de MASON y de la programación de eventos discretos

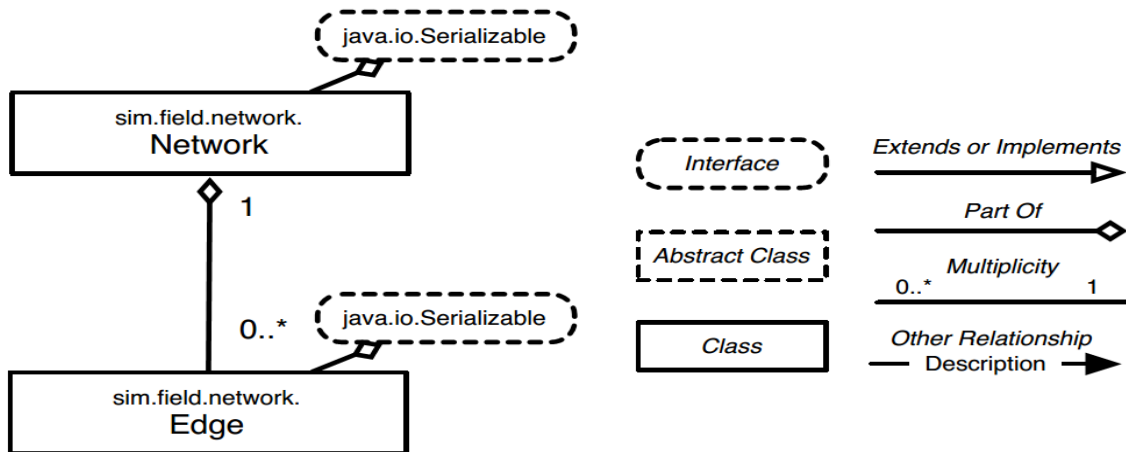


FIGURA 19: Diagrama del paquete Network de MASON

Network: La Figura 19, muestra el paquete `sim.field.network`, el cual consiste en la clase `sim.field.network.Network`, que contiene el gráfico (o red) y la clase `sim.field.network.Edge`. Este es el paquete principal utilizado para la representación de los GIS en MASON.

Modelo y control: En la Figura 20 se muestra la separación clara del modelo y control.

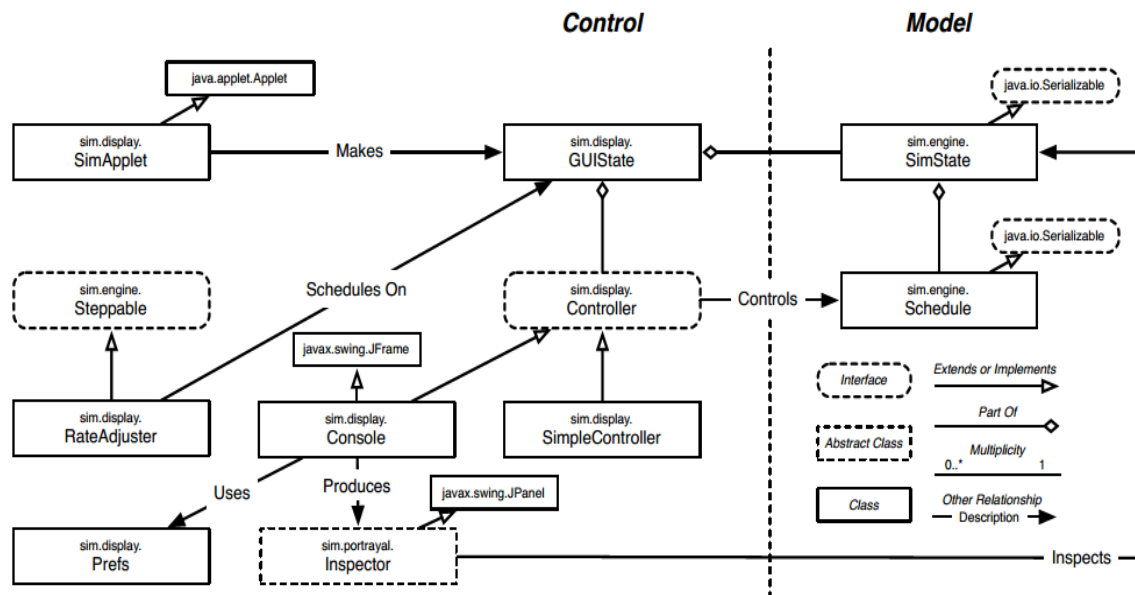


FIGURA 20: Diagrama del código de control de GUI de MASON

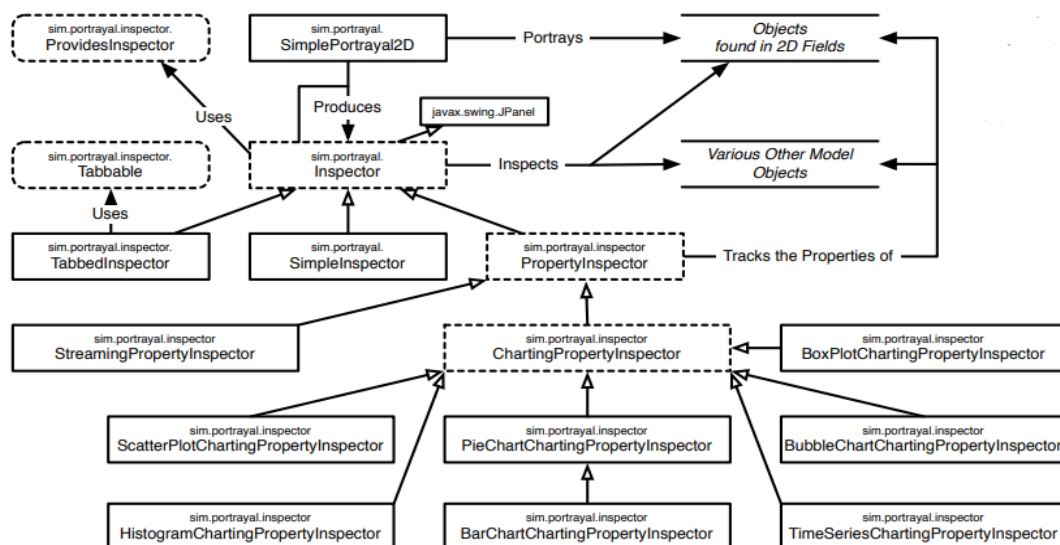


FIGURA 21: Instalación Inspector de MASON

Los inspectores son widget GUI que permiten al usuario inspeccionar, rastrear, trazar un gráfico y modificar un objeto de modelo, un valor específico o la propiedad de un objeto.

Los inspectores son subclases de la clase `sim.portrayal.Inspector`, y con la excepción de `sim.portrayal.SimpleInspector`, todos ellos se encuentran en el paquete `sim.portrayal.inspector`. La Figura 21, muestra un diagrama de las clases de Inspector principales. Los inspectores son generalmente producidos por los aparatos o por otros Inspectores, y su trabajo es inspeccionar los objetos en el modelo según sea necesario [55].

2.4. Casos de Uso del Sistema

Los Casos de Uso del Sistema (CUS) establecen un acuerdo entre clientes y desarrolladores sobre las condiciones y requisitos que debe cumplir el sistema. Es un artefacto narrativo que describe, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario [54]. A continuación, se documentan los artefactos relacionados con los Caso de Uso del Sistema identificados para esta herramienta.

2.4.1. Actores del Sistema

Un actor del sistema es una entidad externa relacionada al sistema y que demanda una funcionalidad al mismo. Esto incluye a los operadores humanos, pero también incluye a todos los sistemas externos. Como bien se ha mostrado, el analista es el único actor del sistema. De este se realiza una breve descripción contenida en la Tabla 9 [7].

TABLA 9: Descripción de los Actores del Sistema

Clases	Descripción
Analista	Es el encargado de iniciar todos los casos de uso del sistema. Tiene la responsabilidad de configurar y ejecutar la simulación, así como obtener los datos estadísticos para posterior análisis. No requiere de estar dotado de conocimientos precisos de informática, más bien basta con un conocimiento básico de operador de aplicaciones.

2.4.2. Diagrama de Casos de Uso del Sistema

A continuación, se muestra en la Figura 22 el Diagrama de Casos de Uso del Sistema, donde quedan representados los casos de uso del sistema definidos y el actor que los inicia. Es necesario destacar que en este caso el actor del sistema deberá obligatoriamente transitar por todos los casos de uso si quiere llegar a exportar los datos debido a la relación de inclusión entre los CUS.

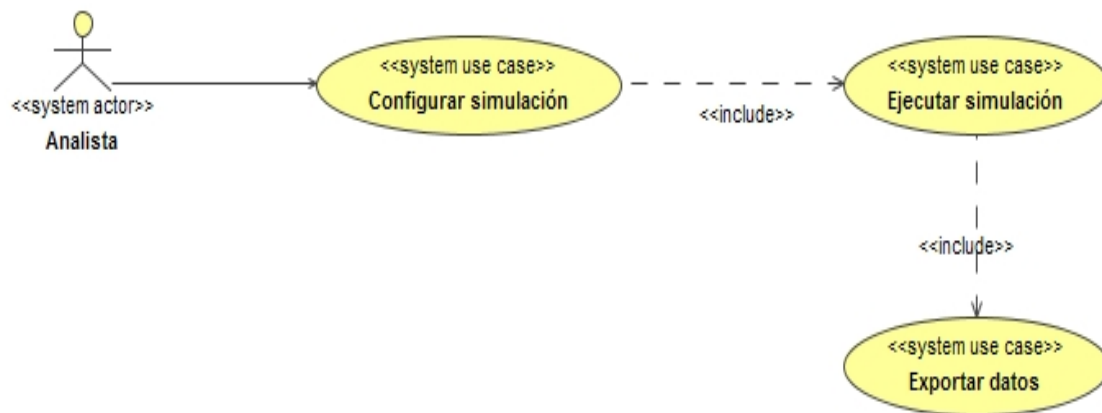


FIGURA 22: Diagrama de Casos de Uso del Sistema

2.4.3. Especificación de los Casos de Uso del Sistema

Las Tablas 10, 11 y 12 muestran la descripción de alto nivel de los casos de uso del sistema.

TABLA 10: Descripción de alto nivel del Caso de Uso “Configurar simulación”

Caso de Uso	Configurar simulación
Actor	Analista
Descripción	Consiste en que el usuario rellene un formulario con los datos de configuración que van a regir el proceso de simulación, como, por ejemplo, cantidad de autos, flujo de los autos etc.

TABLA 11: Descripción de alto nivel del Caso de Uso “Ejecutar simulación”

Caso de Uso	Ejecutar simulación
Actor	Analista
Descripción	Comprende el procedimiento que comienza cuando el usuario inicia la simulación y son representados en ella los vehículos, su interacción, etc. El usuario puede ejecutar la simulación a la velocidad que desea, pausarla o detenerla. Concluye con la finalización de la simulación.

TABLA 12: Descripción de alto nivel del Caso de Uso “Exportar datos”

Caso de Uso	Exportar datos
Actor	Analista
Descripción	Son generados de manera automática datos estadísticos pertenecientes a la simulación, que son útiles para el análisis. El actor no influye en la exportación pero es quien inicia el proceso de ejecución que tiene como resultado este caso de uso.

2.5. Vista de la arquitectura

El patrón arquitectónico de Capas ayuda a estructurar las aplicaciones que pueden ser descompuestas en grupos de subtareas en las que cada grupo de subtareas se encuentra a un nivel particular de abstracción. Se basa en una distribución jerárquica de los roles y las responsabilidades para proporcionar una división efectiva de los problemas a resolver. Los roles indican el tipo y la forma de la interacción con otras capas, y las responsabilidades y la funcionalidad que implementan [53].

Debido a que el sistema descrito puede ser extendido funcionalmente en el futuro, se modela su arquitectura cumpliendo el enfoque por reutilización cuyo diagrama se muestra en la Figura 23.

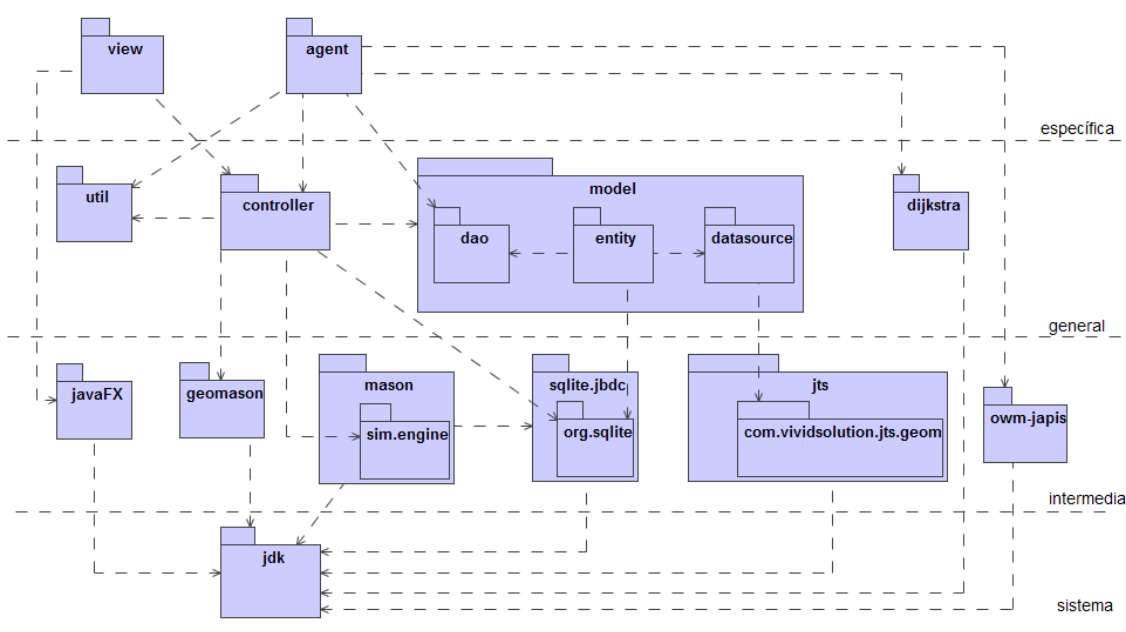


FIGURA 23: Diagrama de estructuración en capas. Enfoque por reutilización

La capa específica contiene los paquetes que agrupan clases propias y muy específicas del sistema, las cuales no se suelen reutilizar. En la capa general se ubican los paquetes que tienen algún grado de reutilización o que requieren de mínimos cambios para ser reutilizados. En la capa intermedia se agrupan los paquetes que poseen el mayor nivel de reutilización, que suelen proveer los marcos de trabajo o bibliotecas que son desarrollados por terceros para ser reutilizados cuando sea necesario.

2.5.1. Descripción de los paquetes

La Tabla 13 contiene la descripción de los paquetes que se muestran en el diagrama de estructuración en capas antes presentado, explicando que tipo de clases encierra cada paquete y la función que realizan las mismas.

TABLA 13: Descripción de los paquetes de la arquitectura

Paquete	Descripción
view	Contiene las clases visuales con las que interactúa el analista.
agent	Lo componen las clases de los agentes y otras entidades que interactúan en la simulación.
controller	En este paquete se encuentran las clases que representan el entorno y otros controladores de los modelos.
util	Está conformado por clases que contienen funcionalidades que facilitan utilidades varias utilidades.
dijkstra	Contiene las clases implementadoras del algoritmo de Dijkstra para el movimiento de los vehículos.
model	Todo el acceso a datos está contenido en este paquete.
mason	Agrupar todos los elementos propios de MASON.
geomason	Contiene la extensión de MASON para el trabajo GIS.
jts	Biblioteca que permite representar geometrías (puntos, líneas, polígonos).
sqlite-jdbc	Biblioteca para la conexión con bases de datos SQLite.
javafx	Contiene los componentes visuales utilizados para la interfaz de usuario.
owm-japis	Contiene las clases de la API de OpenWeatherMap para la consulta del pronóstico del tiempo.

2.6. Clases Principales

A continuación, se da una descripción orientada a las principales clases que forman parte de la solución. Los agentes “semáforo”, “vehículo”, “conductor” y los agentes que representan a factores externos como el ambiente, el pavimento y los obstáculos en la vía constituyen los elementos principales de la simulación y se documentan con sus estructuras y relaciones con el resto de los agentes y clases de la solución.

2.6.1. Agentes de señalización

Una de las entidades con mayor importancia dentro de esta herramienta es el agente semáforo. Sus relaciones con las demás entidades se ilustran en la Figura 24. El semáforo tiene un ciclo, que consiste en una lista de fases. Los vehículos para saber si pueden continuar su tránsito o deben detenerse consultan la fase en la que están transitando y actúan según el indicador de la misma (color de la luz del semáforo). También tiene asociada la geometría en la que está situada, en este caso, un nodo. Cada fase tiene una lista de arcos de entrada, y por cada uno de estos, una lista de arcos de salida. Con esto se logra representar, dada una fase habilitada, qué calles están disponibles para la entrada, y dependiendo de esta, para la salida.

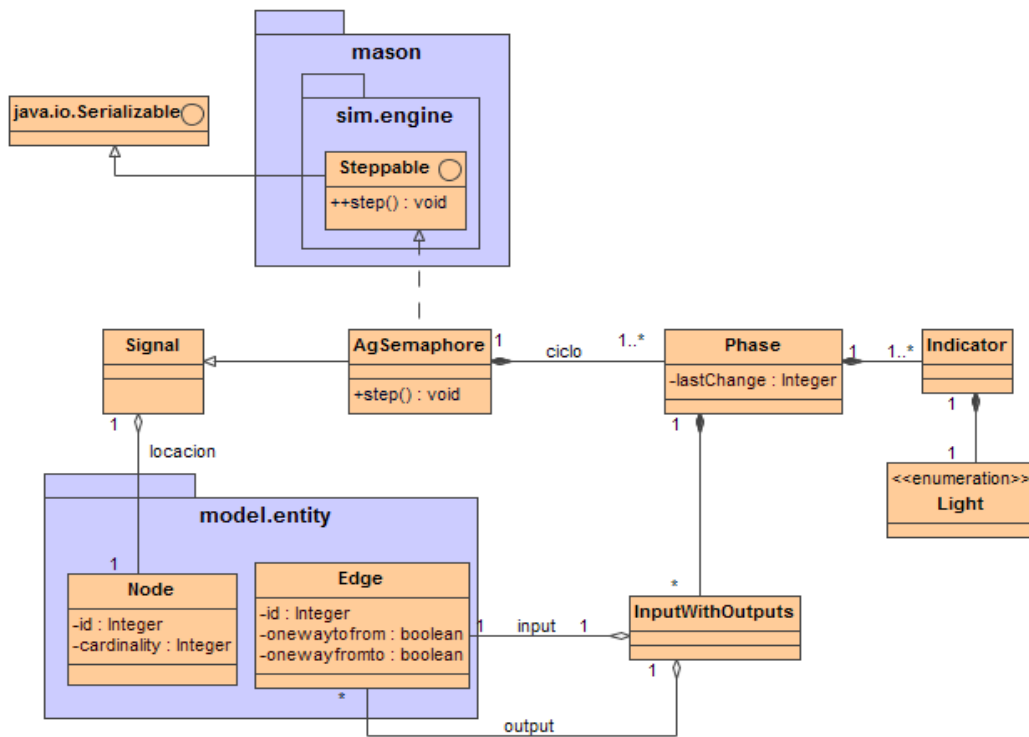


FIGURA 24: Diagrama de Clases relacionadas con el Agente Semáforo

Las otras señales que puedan añadirse a la simulación no tienen ningún comportamiento autónomo por lo que solo serán colocadas por el analista para observar la respuesta de los vehículos ante ellas. La única operación que realizan es validar si pueden ser colocadas en el lugar indicado por el usuario siguiendo las reglas del negocio para instalar las señales planteadas al inicio de este capítulo.

2.6.2. Agente Vehículo

La otra entidad principal de la herramienta desarrollada son los vehículos. Estos constituyen agentes autónomos con una ruta definida desde el inicio de su trayectoria, creada utilizando el algoritmo de Dijkstra [56] para encontrar camino más corto desde el nodo de arribo del vehículo hasta uno de los nodos terminales del entorno de simulación.

Durante la marcha los vehículos comprueban que la calle por la que transitan tenga alguna señal, ya sea semáforo o señales verticales u horizontales, y la almacenan para decidir qué acción tomar cuando lleguen a la línea de señalización. Los vehículos son conducidos por conductores, y las características de este, sumado al tamaño y color del vehículo definen mediante eventos probabilísticos las acciones realizadas por el vehículo. La Figura 25 muestra la relación del agente vehículo con el resto de las clases.

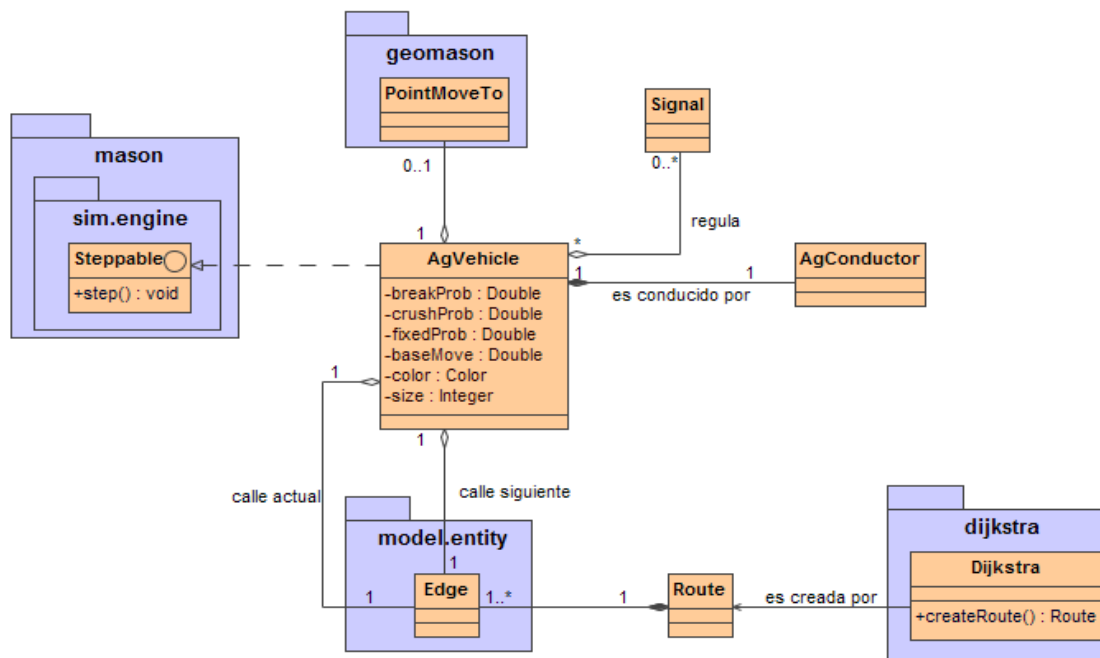


FIGURA 25: Diagrama de Clases relacionadas con el Agente Vehículo

2.6.3. Agente Conductor

El agente conductor es creado para tener en cuenta las características de los usuarios de la vía en la simulación. La clase controladora de la simulación (TrafficSimGeo), que es la contenedora de los vehículos y estos a su vez son los que crean a sus respectivos conductores. Utilizando el atributo “random” de esta clase se crean conductores basándose en valores aleatorios para la edad, sexo, experiencia en la vía, estado anímico en el momento de la simulación y un nivel de distracción que podrá influir en el comportamiento del vehículo que conduce. En la Figura 26 se ilustra la relación entre las clases que interactúan con el conductor.

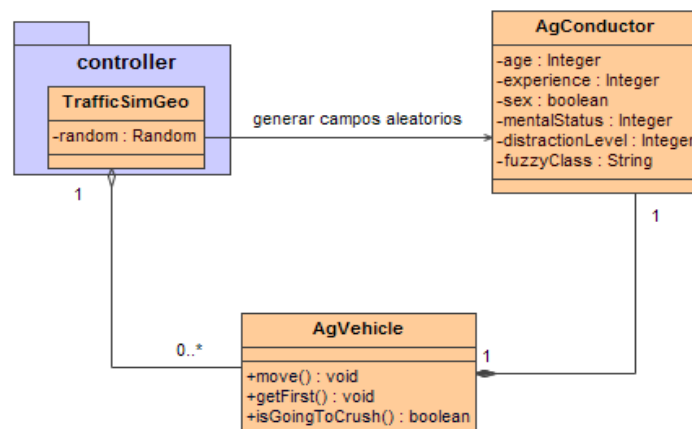


FIGURA 26: Diagrama de Clases relacionadas con el Agente Conductor

Luego de obtenidos estos valores los conductores son agrupados en conjuntos difusos según la edad y la experticia para establecer un comportamiento determinado para cada conjunto.

2.6.4. Agentes que modelan los factores externos presentes en la simulación

Los factores externos a las condiciones del vehículo también son muy influyentes en el resultado de la simulación porque afectan directamente el comportamiento de los mismos. A continuación, se explica cómo funcionan los agentes que modelan estos factores.

Agente Pavimento

El agente pavimento utiliza las probabilidades de que las calles se encuentren en mal estado atendiendo al municipio y al tipo de calle (principal, secundaria, autopista, etc.). Además de estos valores utiliza un valor aleatorio generado por la clase controladora del entorno; la Figura 27 muestra estas relaciones. Para cada calle del entorno es generado un valor entre 0 y 1, y es asignada a esa calle un número entre 1 y 3 según el cuartil en el que se encuentre el valor de probabilidad obtenido.

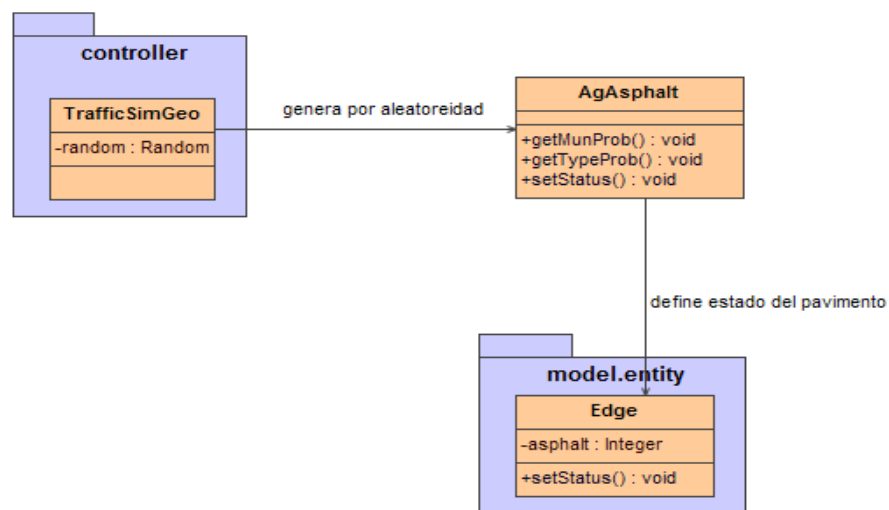


FIGURA 27: Diagrama de Clases relacionadas con el Agente Pavimento

Agente Ambiente

Las condiciones ambientales tenidas en cuenta para la simulación son la temperatura, la humedad, el momento del día simulado y la lluvia. En caso de que el usuario ejecute la simulación en línea el agente ambiente obtendrá el pronóstico del tiempo actual utilizando la API de OpenWeatherMap [57].

La actualización de las variables observadas es mostrada por el entorno en cada paso de la simulación y los vehículos reciben la notificación correspondiente para que moderen su velocidad teniendo en cuenta el estado ambiental en todo momento.

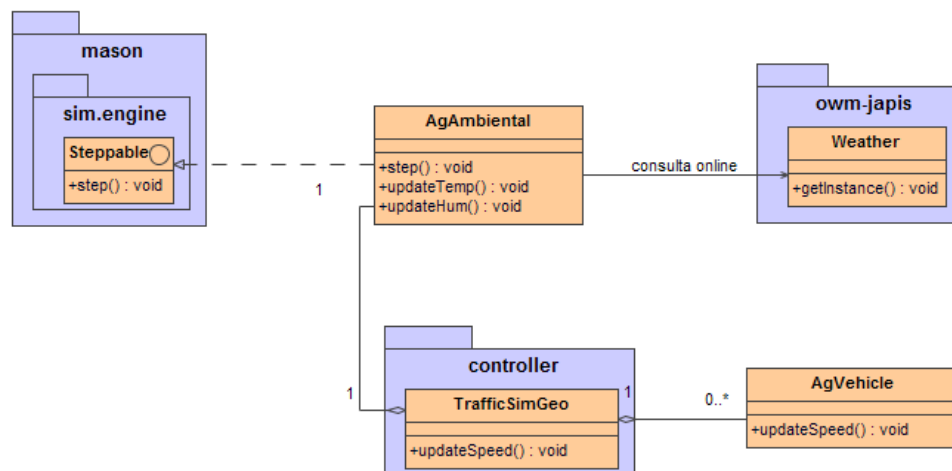


FIGURA 28: Diagrama de Clases relacionadas con el Agente Ambiente

2.7. Patrones y principios de diseño

Los patrones de diseño son descripciones de objetos y clases comunicantes que son ajustadas a resolver un problema general de diseño. Definen micro arquitecturas de subsistemas de componentes. Resuelven problemas de diseño que permiten darle claridad al diseño, multiplicación de clases y adaptabilidad a requerimientos cambiantes.

A continuación, se documentan los patrones de diseño utilizados en la solución propuesta y los principios de diseño asociados a cada uno de ellos.

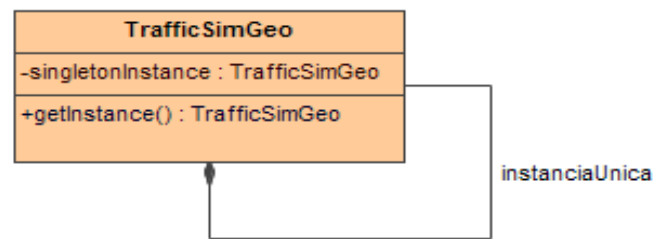
2.7.1. Patrón Singleton

Este constituye un patrón creacional que asegura que una clase posea una sola instancia y provee un único punto de acceso a la misma. Para esto, dicha clase es responsable de guardar su única instancia, de esta manera se asegura de que ninguna otra instancia de ella pueda ser creada. Imposibilita las peticiones para crear nuevos objetos de su tipo, proporcionando una única vía para acceder a la dicha instancia [58].

La propuesta de solución incluye varias clases que cumplen este patrón. Una de estas es la clase “TrafficSimGeo”.

En la Figura 29 se puede observar que esta clase posee una instancia de sí misma llamada “singletonInstance”, la cual es estática. Solo posee un constructor, y este es privado. Tiene un método “getInstance()” que es el único que invoca al constructor, y es la única forma de obtener la instancia de esta clase.

Otras de las clases que implementan este patrón son “MainController” y “AgStatistical” y se representan de la misma manera.

FIGURA 29: Patrón *Singleton*, Clase TrafficSimGeo

Con la aplicación de este patrón se garantizan los siguientes **principios de diseño**:

- *Única responsabilidad*: El método “getInstance()” es la única forma de crear y obtener una instancia de “TrafficSimGeo”, evitando que se creen más de una, es decir, es la única forma de acceder al constructor.
- *Principio Hollywood*: La clase “TrafficSimGeo” es de alto nivel y con el método “getInstance()” devuelve el único objeto, evitando que las clases de más bajo nivel invoque su constructor.

2.7.2. Patrón Facade

El paquete “util” contiene una variedad de funcionalidades implementadas que fueron necesarias utilizar en la solución desarrollada. Pese a que no forman parte directamente de la lógica de la solución, las funcionalidades del paquete “util” constituyen dependencias de esta lógica. Una variedad de paquetes hace uso de las funcionalidades del paquete “util”, en principio, esto provoca un acceso desordenado, lo que hace más complicado el modelado. Esto se evidencia en la primera iteración del diseño de estas utilidades, la cual se puede observar en la Figura 30.

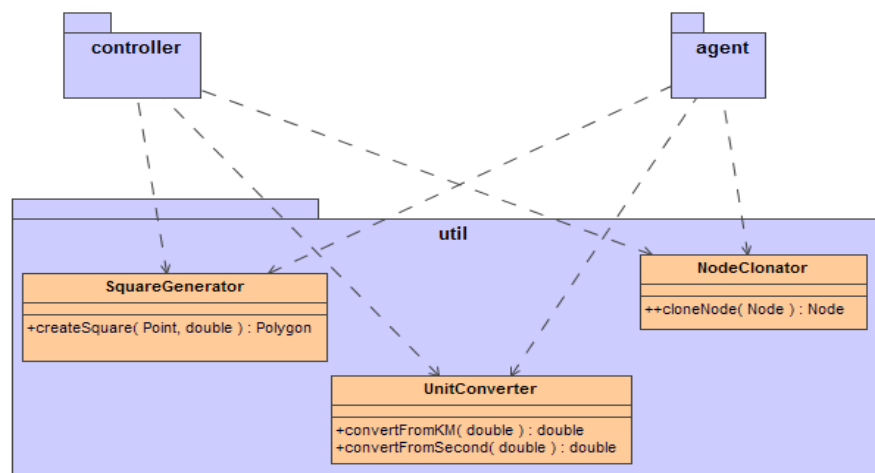


FIGURA 30: Diseño de las utilidades, iteración 1

Con propósito de eliminar este problema frecuente se decide hacer uso del patrón *Facade* [7]. Al patrón *Facade* (Fachada en español), se le da uso con el propósito de proporcionar una interfaz unificada de alto nivel que, representando a todo un subsistema, facilite su uso. La “fachada” satisface a la mayoría de los clientes, sin ocultar las funciones de menor nivel a aquellos que necesiten acceder a ellas [59].

Para su aplicación, se crea una nueva clase abstracta llamada “FacadeOfTool” que funciona como intermediaria entre los componentes del paquete “util” y los paquetes y subsistemas externos a este. En la Figura 31 se ilustran las modificaciones mencionadas.

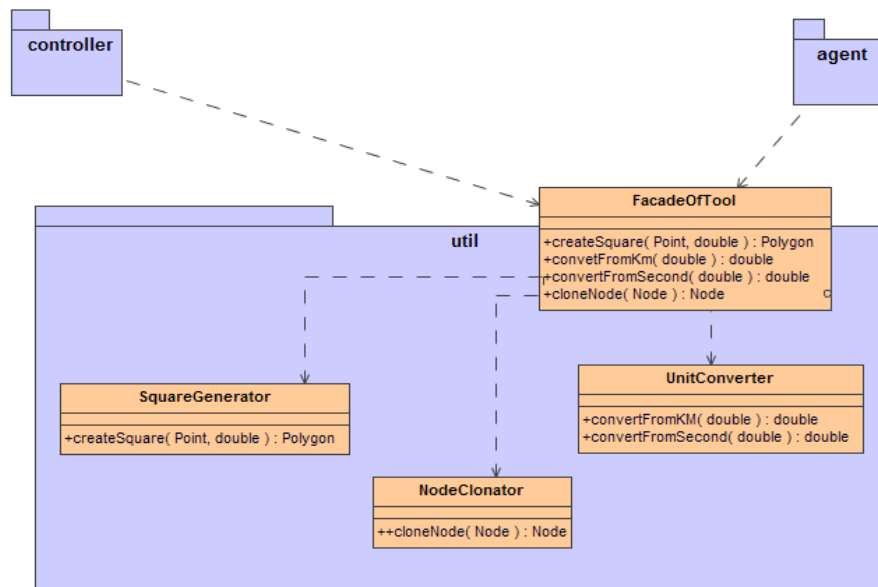


FIGURA 31: Diseño de las utilidades, iteración 2 (Patrón *Facade*)

Junto a la aplicación de este patrón se cumplen los siguientes **principios de diseño**:

- *Principio Hollywood*: Se ha diseñado una pieza de alto nivel (FacadeOfTool) que interactúa con piezas de más bajo nivel (resto de las contenidas en el paquete “util”) haciendo el papel de coordinadora.
- *Bajo acoplamiento*: Solo hay una clase acoplada (FacadeOfTool) por lo que simplifica y centraliza un conjunto de invocaciones en un único punto; proporcionando un acoplamiento débil entre el paquete “util” y los que dependen de este.
- *Ley de Demeter*: Las clases cliente se relacionan únicamente con la clase “FacadeOfTool” minimizando el impacto de los cambios y encapsulando en la Fachada las particularidades.

2.7.3. Patrón Observer

El objetivo de este patrón es definir una dependencia uno a muchos entre objetos de forma de que, cuando el estado de uno de ellos cambia, todos los objetos dependientes son notificados y actualizados de forma automática [58]. Este comportamiento es útil para que los vehículos sepan cuando modificar su velocidad en dependencia del estado ambiental o de las condiciones del pavimento por el que transitarán. La relación entre las clases de la solución que modelan este patrón se muestra en la Figura 32.

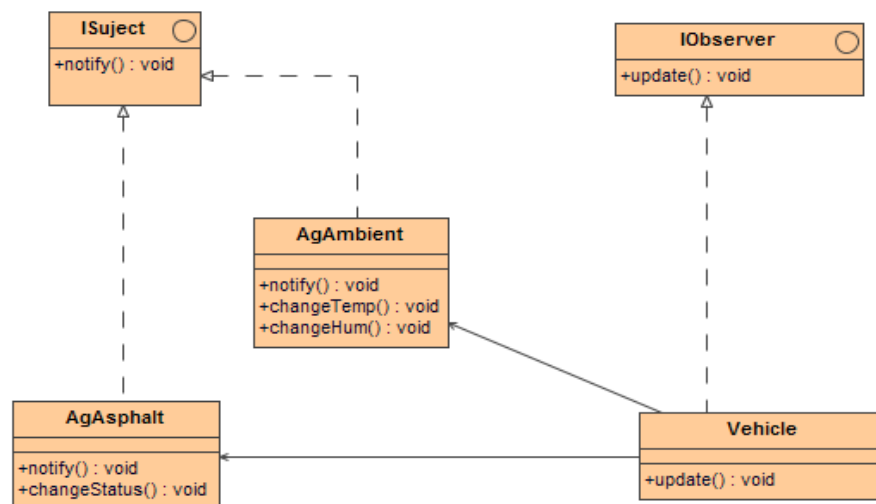


FIGURA 32: Patrón Observer

Con la implementación de este patrón se cumplen con los siguientes **principios de diseño**:

- *Ley de Demeter*: El sujeto realiza llamados al observador (método Update) porque tiene una relación directa con él.
- *Encapsular la variabilidad*: Se identifica como método que cambia el método Update que implementa lo que el observador hará cuando cambia algo en el sujeto.
- *Bajo acomplamiento*: El sujeto y el observador interactúan entre ellos, pero apenas tienen conocimiento el uno sobre el otro.
- *Open- Close*: Cerrado para modificaciones por los métodos de adicionar y eliminar observadores y notificarles un cambio, pero abierto porque cada sujeto tiene variables diferentes sobre las que hay que controlar el estado.

2.7.4. Patrón DAO

El patrón de acceso a datos (DAO, *Data Access Object*) tiene como objetivo desacoplar el acceso a datos de su subyacente almacenamiento. Este patrón propone el uso de interfaces (denominados

objetos de acceso a datos) como mecanismo para comunicar los componentes de la aplicación con los dispositivos de almacenamiento de datos (tales como bases de datos o ficheros). A través de este mecanismo se logra desacoplar la implementación de la fuente de datos utilizada.

La Figura 33 muestra cómo se le dio uso a este patrón. Primeramente, se encuentran los objetos encargados de mapear la información proveniente de los datos, en este caso, base de datos relacional “SQLite”. Luego están las “Interfaces DAO” que agrupan la esencia de la arquitectura de este patrón. La interfaz DAO contiene los principales métodos como la búsqueda por id y búsqueda total. Las interfaces EdgeDAO, NodeDAO, StatisticalDAO, SignalDAO, AccidentDAO y ConductorDAO se encargan de redefinir los métodos que heredan de DAO y definir nuevas funcionalidades para los modelos Edge, Node, Statistical, Signal, Accident y Conductor respectivamente. Por último, están las implementaciones de las interfaces anteriormente descritas.

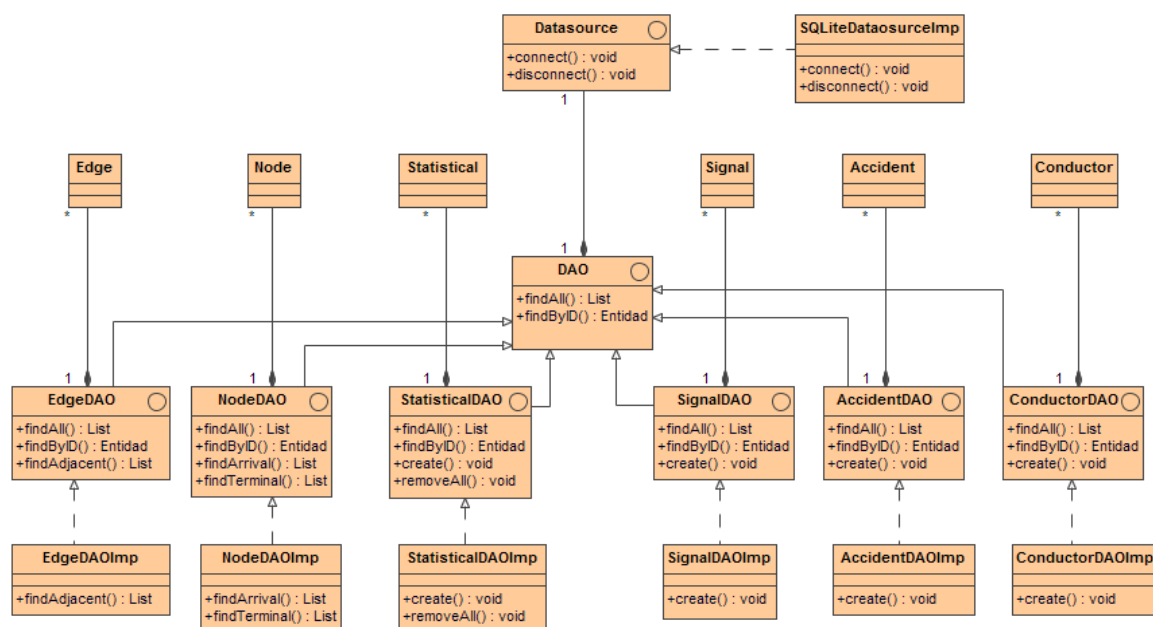


FIGURA 33: Patrón DAO

2.8. Modelo de datos

En el desarrollo de esta herramienta se utilizan dos bases de datos (DB, *database*) con sistema SQLite [31]. La primera tiene como objetivo almacenar los datos espaciales y la segunda, los datos correspondientes a los resultados estadísticos como cantidad de accidentes, tiempo de espera promedio y número de infracciones que resultan de la ejecución de cada simulación.

2.8.1. Datos espaciales

Para el almacenamiento y consulta de la información espacial se utilizó una base de datos SQLite en conjunto con su extensión SpatiaLite [32]. Para generar esta DB se utilizó la herramienta “spatialite_osm_net”, que forma parte del conjunto de herramientas que brinda el proyecto SpatiaLite para el trabajo con ficheros de OSM. A través del uso de dicha solución es posible generar un grafo partiendo de las etiquetas y relaciones presentes en la información de OSM que puede ser utilizado posteriormente para el enrutamiento mediante VirtualNetwork de SpatiaLite.

Modelo Lógico

La Figura 34 muestra la relación entre las entidades que forman parte del modelo lógico de la base de datos espacial, con los atributos que poseen y que permiten a los agentes obtener información de la red espacial para poder transitar sobre ella.

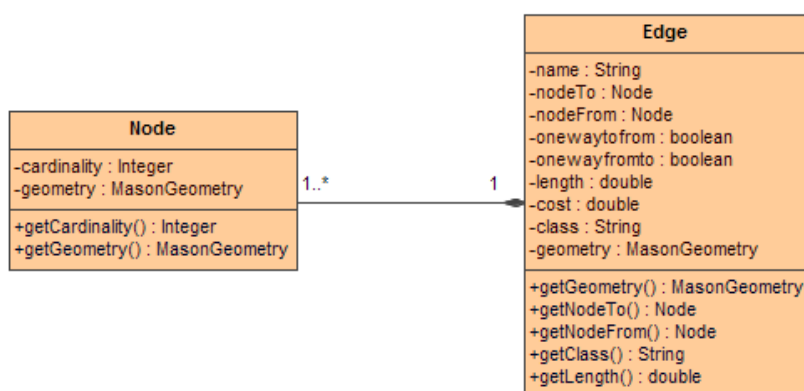


FIGURA 34: Modelo lógico de la Base de Datos Espacial

Modelo físico

El esquema resultante del uso de la herramienta “spatialite_osm_net” se muestra en la Figura 35. La herramienta “spatialite_osm_net” no genera llaves foráneas (FK, *Foreign Key*), no obstante, está documentado en su guía de uso [9] que los atributos “node_from” y “node_to” de la tabla “roads”, se hacen corresponder con el atributo “node_id” de la tabla “roads_nodes” respectivamente.

En la Tabla 14 y la Tabla 15 se muestra la descripción de los atributos de esta base de datos. La tabla “roads” contiene la información de las calles y avenidas, y la tabla “roads nodes” contiene los datos de los puntos extremos de estas líneas.

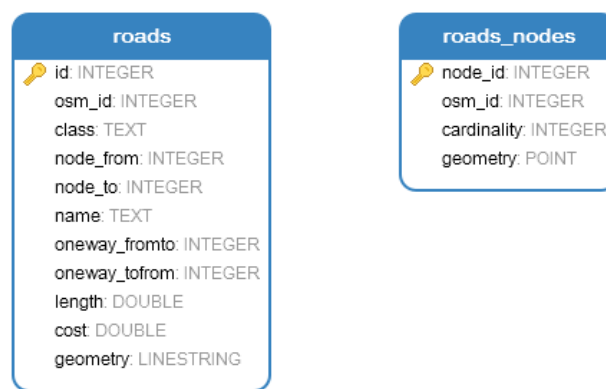


FIGURA 35: Modelo físico de las tablas

TABLA 14: Descripción de los atributos de la tabla “roads”

Atributo	Descripción
id	Este elemento es la llave primaria (PK, <i>primary key</i>) de la tabla y, por lo tanto, el identificador de cada tupla.
name	Nombre de la calle.
node from	Identificador del nodo de origen.
node to	Identificador del nodo destino.
oneway fromto	La calle tiene sentido habilitado del nodo origen al nodo destino.
oneway tofrom	La calle tiene sentido habilitado del nodo destino al nodo origen.
class	Tipo de calle, puede ser una calle primaria, secundaria, autopista, entre otras.
length	Longitud de la calle medida en Index (unidad de la simulación).
cost	Costo para los vehículos al transitar por esa calle.
geometry	Dato binario codificado por Spatialite. Contiene las coordenadas de todos los puntos que forman la línea.

TABLA 15: Descripción de los atributos de la tabla “roads nodes”

Atributo	Descripción
id	Indica la cardinalidad, es decir, la cantidad de líneas que empiezan o terminan en este punto.
cardinality	Indica la cardinalidad, es decir, la cantidad de líneas que empiezan o terminan en este punto.
geometry	Dato binario codificado por Spatialite. Contiene las coordenadas del punto.

2.8.2. Datos estadísticos

La segunda base de datos almacena los datos correspondientes a los resultados estadísticos como cantidad de accidentes, tiempo de espera promedio y número de infracciones que resultan de la ejecución de cada simulación.

Modelo lógico

La Figura 36 ilustra cómo es la relación entre las entidades que conforman el modelo lógico de la base de datos estadística. Las entidades Conductor, Signal y Accident son creadas para registrar cada uno de las instancias respectivas y poder enfocar la toma de decisiones en elementos concretos sobre los accidentes, las señales y los conductores involucrados.

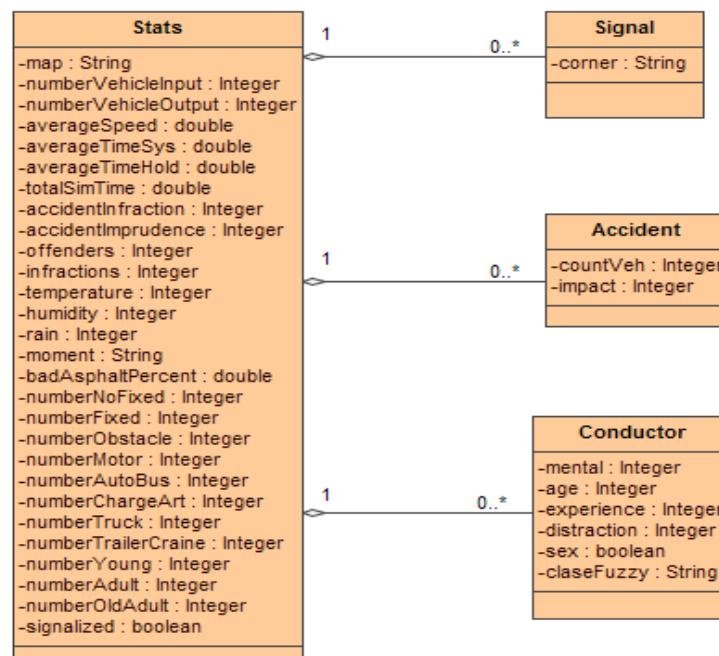


FIGURA 36: Modelo lógico de las tablas

Modelo físico

La Figura 37 muestra el modelo físico de la base de datos estadística. Esta base de datos fue creada durante el desarrollo de la herramienta por lo que sí presenta llaves foráneas en las relaciones de “uno a mucho” representadas. Las Tablas 16, 17, 18, 19 y 20 se encuentran las descripciones de cada uno de los atributos de las tablas que conforman esta base de datos.

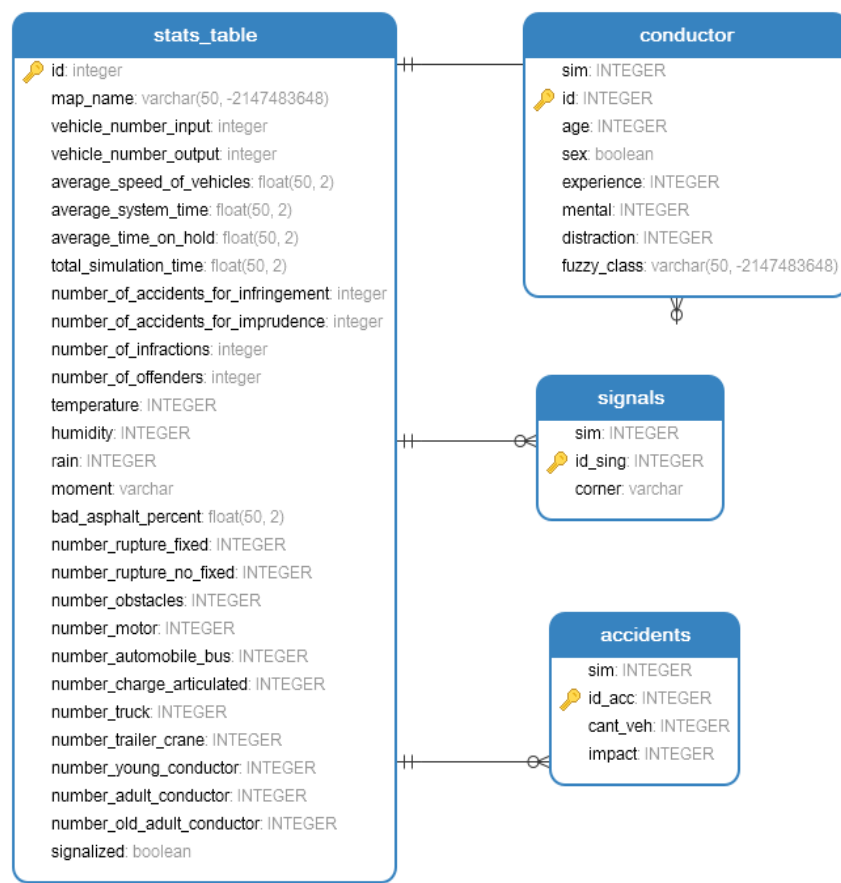


FIGURA 37: Modelo físico de las tablas

TABLA 16: Descripción de los atributos de la tabla “stats” (1)

Atributo	Descripción
id	PK de la tabla.
map name	Cantidad de vehículos arribados.
vehicle number input	Cantidad de vehículos que salen del sistema.
vehicle number output	Cantidad de vehículos que salen del sistema.
average speed of vehicles	Velocidad promedio de los vehículos.
average system time	Tiempo promedio en el sistema.
average time on hold	Tiempo de espera promedio.
total simulation time	Tiempo total de simulación.
number of accidents for infringement	Número de accidentes por infracción.
number of accidents for imprudence	Número de accidentes por imprudencia.
number of infractions	Número de infracciones.
number of offenders	Número de infractores.
temperature	temperatura promedio de la simulación.

TABLA 17: Descripción de los atributos de la tabla “stats” (2)

Atributo	Descripción
humidity	Humedad promedio de la simulación.
rain	Lluvia promedio de la simulación.
moment	Momento del día simulado.
bad asphalt percent	Porcentaje de calles en mal estado simuladas.
bad asphalt percent	Porcentaje de calles en mal estado simuladas.
number rupture fixed	Número de vehículos arreglados.
number ruoture no fixed	Número de vehículos no arreglados.
number of obstacles	Número de obstáculos simulados.
number of motos	Número de ciclos simulados.
number of automobile/bus	Número de automóviles o autobuses.
number of charge/articulated	Número de vehículos de carga o articulados.
number of truck	Número de tractores simulados.
number of trailer/crane	Número de remolques o grúas.
number of young conductor	Número de conductores jóvenes.
number of adult conductor	Número de conductores adultos.
number of old adult conductor	Número de conductores adultos mayores.
Signalized	Verdadero si existen semáforos.

TABLA 18: Descripción de los atributos de la tabla “conductor”

Atributo	Descripción
id	PK de la tabla.
id sim	FK, id de la simulación a la que pertenece.
age	Edad del conductor.
experience	Cantidad de años de experiencia conduciendo.
mental status	Estado de ánimo que presenta el conductor.
distraction level	Nivel de distracción del conductor.
sex	Sexo del conductor. Verdadero si es mujer.
fuzzy class	Clasificación difusa del conductor según su edad.

TABLA 19: Descripción de los atributos de la tabla “signal”

Atributo	Descripción
id	PK de la tabla.
id sim	FK, id de la simulación a la que pertenece.
corner	Esquina donde se encuentra la señal.

TABLA 20: Descripción de los atributos de la tabla “accident”

Atributo	Descripción
id	PK de la tabla.
id sim	FK, id de la simulación a la que pertenece.
cant veh	Cantidad de vehículos involucrados en el accidente.
impact	Impacto del accidente según la cantidad de vehículos y sus tamaños.

2.9. Interfaz de usuario

Para la interfaz de usuario fueron tenidas en cuenta varias metas de diseño que, en su mayoría están orientadas a la usabilidad, y en una menor parte, a la experiencia del usuario. Las metas de usabilidad tienen como objetivo diseñar un sistema eficaz que les permitirá a los usuarios ser muy productivos en su trabajo. Siguen normas orientadas a cuestiones éticas como, uso de lenguaje técnico y profesional, ausencia de palabras vulgares y omisión de insultos. Como se muestra en la Figura 38 los principios básicos en el diseño de las pantallas fueron:

- La mayoría de las pantallas son de no menos de 800 píxeles por 600 píxeles.
- Las ventanas y diálogos coinciden con el flujo de trabajo principal del usuario.
- Se usa una pantalla “Home” como base.
- No se utiliza el color sólo para fines estéticos y no se referencia por colores.
- Se utilizan componentes visuales que minimicen los errores de los usuarios como los “combobox”.



FIGURA 38: Interfaz de usuario

Las acciones del usuario están validadas al igual que los datos de entrada. Cuando el usuario comete un error es notificado por el sistema, informando el lugar en el que ha cometido y la manera de solucionarlo. Como parte de la retroalimentación al usuario, se implementan vistas indicando el estado del sistema. De la misma forma se envían de notificaciones por parte del sistema al usuario cuando cierto evento finaliza, indicando el resultado de este. La Figura 39 muestra las notificaciones del sistema en varias situaciones.



FIGURA 39: Notificaciones del sistema

El usuario tiene en todo momento control sobre el sistema pudiendo salir o volver a la pantalla anterior en cualquier punto donde se encuentre. Esto se muestra en la Figura 40, junto con los componentes de control usados para disminuir la entrada de errores por parte del usuario.

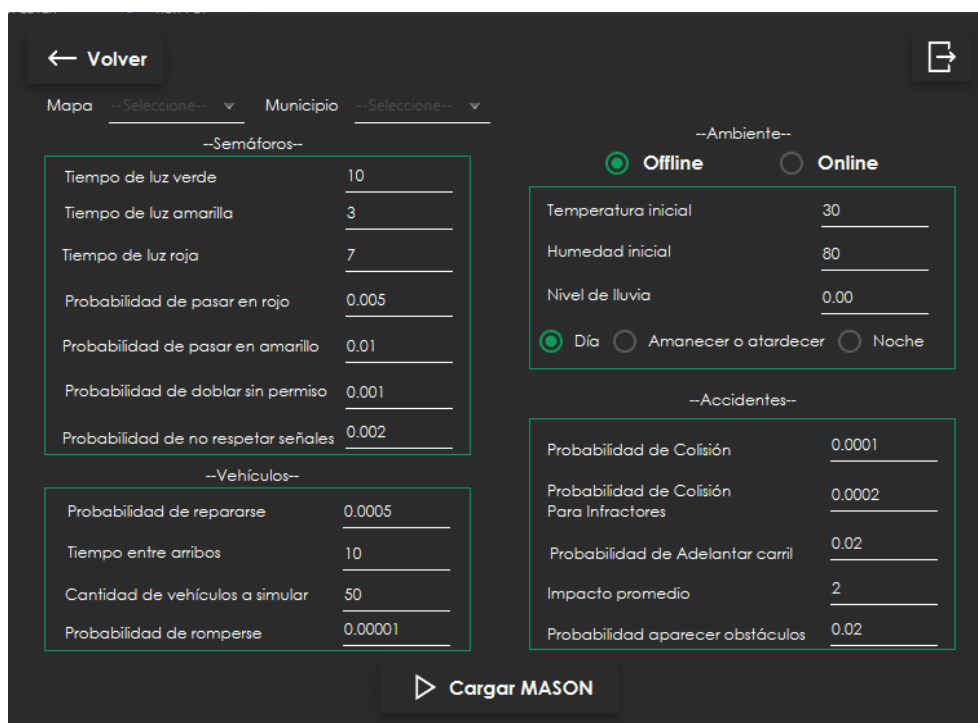


FIGURA 40: Interfaz de usuario

La interfaz de usuario desarrollada también permite navegar directamente por la página de OpenStreetMap en caso de que el usuario desee descargar los mapas en el momento en que ejecuta el sistema. La Figura 41 muestra la pantalla del sistema donde se puede seleccionar buscar los mapas en línea o usar los que estén almacenados en alguna ubicación del dispositivo.



FIGURA 41: Variantes para obtener mapas

2.10. Conclusiones parciales

Concluido el capítulo de la propuesta de solución se arriban a las conclusiones siguientes:

- Se identificaron los principales elementos de análisis de la solución.
- Se obtuvo una solución que debe ser escalable y reutilizable que puede ser fácilmente adaptada y extendida a nuevas funcionalidades.
- SQLite se utilizó como una alternativa de almacenamiento de datos espaciales eficaz, que con la extensión SpatiaLite, logra brindar muchas facilidades para la obtención de geometrías y cálculo de rutas.
- Al aplicar principios en el diseño de las pantallas, la interacción humano computadora es eficaz.
- Se les agregaron a los vehículos los comportamientos para que actuaran según el estado del ambiente y de las calles.
- Se les agregaron a los vehículos los comportamientos para que pudieran sufrir desperfectos técnicos durante su ruta.
- Se añadieron a los vehículos comportamientos para que actúen sus características y las de su conductor.

CAPÍTULO 3

Validación de la Herramienta

En el presente capítulo se presenta la validación de la herramienta desarrollada. Para ello se realizan un conjunto de pruebas funcionales, además se establece una comparación entre los resultados estadísticos obtenidos para diferentes juegos de datos. También se realiza un Diseño de Experimentos, donde se pone en uso la herramienta para un conjunto de experimentos a fin de sugerir una configuración semafórica para el caso “Ciudad Deportiva”.

3.1. Pruebas Funcionales

Las pruebas de caja negra, también llamadas pruebas de comportamiento o pruebas funcionales, se enfocan en los requisitos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requisitos funcionales para un programa. Las pruebas de caja negra no son una alternativa para las técnicas de caja blanca. En vez de ello, es un enfoque complementario que es probable que descubra una clase de errores diferente que los métodos de caja blanca [53].

Las pruebas de caja negra intentan encontrar errores en las categorías siguientes:

1. funciones incorrectas o faltantes,
2. errores de interfaz,
3. errores en las estructuras de datos o en el acceso a bases de datos externas,
4. errores de comportamiento o rendimiento y
5. errores de inicialización y terminación.

3.1.1. Guiadas a casos de uso

Al software se le realizaron tres pruebas funcionales guiadas a casos de uso que se muestran en las Tablas 21, 22 y 23.

TABLA 21: Caso de Prueba: Cargar pronóstico del tiempo del servicio web

Caso de Prueba	Cargar pronóstico del tiempo del servicio web
Caso de Uso	Configurar simulación
Desarrollador	Ariadna C. Moreno Román
Curso a probar	Consulta de los datos del estado del tiempo desde un servicio web
Probador	Ariadna C. Moreno Román
Objetivo de la prueba	Comprobar que la simulación obtiene correctamente los datos del servicio web meteorológico
Descripción de la prueba	El usuario selecciona la opción de obtener los datos del ambiente de manera online y ejecuta la simulación
Condiciones	
Condiciones de entrada	Resultados Esperados
Se selecciona la opción de cargar los datos del tiempo online. Es necesario tener conexión a Internet	La simulación inicia correctamente mostrando en la pantalla de Detalles de la Simulación la temperatura, humedad y lluvia obtenidas del servicio web
Observaciones	
Los resultados obtenidos son iguales a los esperados	

TABLA 22: Caso de Prueba: Identificar vehículos infractores y afectados

Caso de Prueba	Identificar vehículos infractores y afectados
Caso de Uso	Ejecutar simulación
Desarrollador	Ariadna C. Moreno Román
Curso a probar	Coloración de la etiqueta de los vehículos según su estado técnico y su comportamiento en la vía
Probador	Ariadna C. Moreno Román
Objetivo de la prueba	Comprobar que cambia de color la etiqueta de los vehículos según su comportamiento durante la simulación
Descripción de la prueba	El usuario inicia la simulación de manera normal.
Condiciones	
Condiciones de entrada	Resultados Esperados
Se coloca un semáforo en alguna intersección	Debe tornarse de color naranja la etiqueta de los vehículos que se lleven la luz roja o amarilla, de color morado la etiqueta de los vehículos que adelantan carril, de color verde la etiqueta de los vehículos rotos y de color rojo la de los vehículos accidentados
Observaciones	
Los resultados obtenidos son iguales a los esperados	

TABLA 23: Caso de Prueba: Detener simulación

Caso de Prueba	Detener simulación
Caso de Uso	Ejecutar simulación
Desarrollador	Ariadna C. Moreno Román
Curso a probar	Detener la simulación luego de un número de autos simulados
Probador	Ariadna C. Moreno Román
Objetivo de la prueba	Verificar que el sistema detiene la simulación cuando se ha llegado al máximo de vehículos configurados para esa simulación
Descripción de la prueba	Se introduce una cantidad de vehículos a simular y se ejecuta la simulación
Condiciones	
Condiciones de entrada	Resultados Esperados
Se introduce como cantidad de vehículos a simular 50	La simulación debe ejecutarse normalmente y detenerse cuando el vehículo número 50 sale del entorno de simulación por uno de los nodos de salida.
Observaciones	
Los resultados obtenidos varían un poco de los esperados. El sistema sigue generando autos después del vehículo 50 pero sí detiene la simulación cuando el vehículo 50 sale del entorno.	

3.1.2. Clases de equivalencia

La técnica de clases de equivalencia se usa con el propósito de identificar los valores de entrada para los cuales se estima que el comportamiento sea similar. De esta forma, se crean clases agrupando a los mismos, y luego se elige un valor de cada clase y se realiza una prueba con él [53]. Así, se estaría logrando una amplia cobertura. Los casos de uso “Configurar simulación” y “Ejecutar simulación” son los que estarán relacionados con las pruebas que se desarrollan bajo esta técnica. La Tabla 24 muestra, los valores elegidos para analizar las clases de equivalencia. Se puede observar que se tomaron valores excesivos, de forma tal que se realicen pruebas orientadas al rendimiento, cuando la herramienta está bajo estrés.

TABLA 24: Clases de equivalencia

Parámetros	Descripción	Valor por debajo del dominio	Normal	Excesivo
VPA	Vehículos por arribos	-1	1	200
CMV	Cantidad máxima de vehículos	-1	50	500
TFS	Tiempo de las fases del semáforo (verde/amarillo/rojo)	-10	10/3/7	150

Las pruebas desarrolladas siguiendo esta técnica están descritas en la Tabla 25.

TABLA 25: Prueba de las clases de equivalencia

Casos de Prueba	Datos insertados	Respuesta del sistema
Test 1 Flujo Principal (Datos correctos)	VPA = 1 CMV = 50 TFS = 10/3/7	La simulación carga correctamente, las configuraciones se comprueban que son las introducidas, la simulación se desarrolla de forma esperada.
Test 2 Valores fuera de rango para las cantidades de vehículos	VPA = -5 CMV = -10 TFS = 10/3/7	El sistema no permite ingresar valores negativos en la configuración de la simulación.
Test 3 Prueba de estrés Valores considerablemente grandes para las cantidades de vehículos	VPA = 100 CMV = 150 TFS = 10/3/7	Permite iniciar la simulación, pero lanza una excepción de acceso concurrente, no se cierra la aplicación y se pueden modificar los valores y volver a iniciarla.
Test 4 Semáforos con contadores negativos	VPA = 1 CMV = 50 TFS = -10/-3/-7	El sistema no permite ingresar valores negativos en la configuración de la simulación.
Test 5 Semáforos con tiempos excesivos	VPA = 1 CMV = 50 TFS = 120/80/90	La simulación carga correctamente, los semáforos respetan los tiempos y se desenvuelven normalmente.

3.1.3. Análisis de los resultados de las pruebas

El diseño y ejecución de los casos de prueba antes descritos, constató la correcta implementación de las funcionalidades presentes en la herramienta, así como la capacidad de esta de responder ante errores y brindar mensajes informativos al usuario. A pesar de que el resultado de la mayoría de las pruebas fue el esperado, no es posible afirmar que la aplicación esté libre de otros errores. Los casos de pruebas deben ser ejecutados en diferentes sistemas operativos, para garantizar un comportamiento similar de la aplicación independiente de la plataforma donde se encuentre instalada. Respecto a las pruebas de rendimiento realizadas se puede concluir que existen fallos en cuanto a accesos concurrentes a la memoria, los tiempos de respuesta obtenidos son considerablemente pequeños y que la aplicación tiene un funcionamiento fluido.

3.2. Comparación de resultados con diferentes configuraciones

En la configuración de la herramienta existen varios parámetros que son decisivos en el comportamiento de los vehículos durante la simulación. Como parte de la validación del software se analizan los resultados estadísticos brindados por la herramienta luego de tres ejecuciones con juegos de datos distintos, la Tabla 26 contiene los parámetros específicos bajo los cuales se ejecutó cada simulación y la Tabla 27 muestra los parámetros comunes para todas.

TABLA 26: Juegos de datos de las ejecuciones

Parámetros	Simulación A	Simulación B	Simulación C
Tiempo de las luces	10/3/7	20/6/14	5/1/4
Tiempo entre arribos	10	1	15
Lluvia	0	5	3
Horario de la simulación	Amanecer	De día	Madrugada
Temperatura	27	30	22
Humedad	65	80	65

TABLA 27: Parámetros comunes para todas las simulaciones

Parámetros	Valor
Mapa	Avenida 124 y Avenida 51
Municipio simulado	Marianao
Probabilidad de pasar en rojo/amarillo	0.0001/0.001
Probabilidad de colisión	0.0001
Probabilidad de adelantar carril	0.02
Probabilidad de romperse/repararse	0.00001/0.0005
Semáforo instalado	Avenida 124 y Avenida 51

Los resultados de los análisis hechos con las estadísticas de las tres ejecuciones se muestran gráficamente en las siguientes figuras. La Figura 42 ilustra la relación entre las cantidades de accidentes, infracciones y roturas ocurridas. Es evidente que en la Simulación B, al existir mayor demanda de vehículos en la vía y ser mayor el tiempo de luz roja del semáforo, la cantidad de infracciones es mayor y con ello aumenta también la cantidad de accidentes. La simulación A transcurre sin inconvenientes debido a que los vehículos esperan un tiempo aceptable por las luces del semáforo, no hay tantos vehículos en la simulación y se realiza en un horario donde los conductores prestan total atención a la vía. Por último, la simulación C también se desarrolla con cierta normalidad, salvo por el hecho de que en horarios de la noche los conductores suelen desobedecer las señales por la falta de personal policial en las calles, y ello se evidencia en el número de infracciones cometidas.

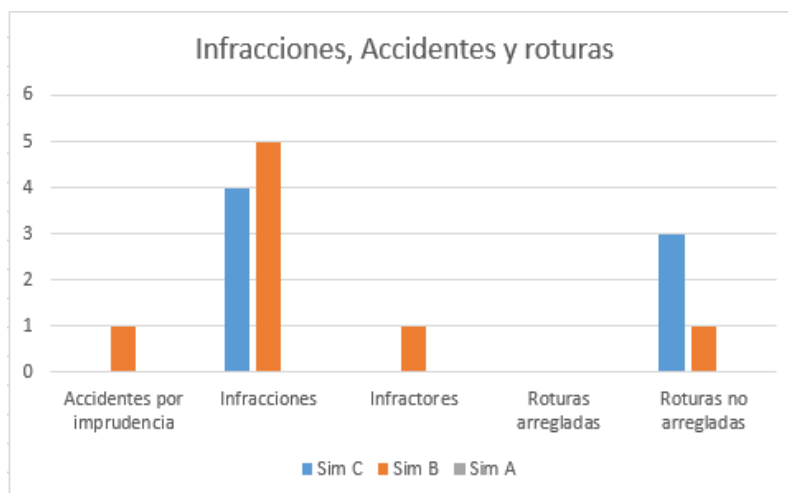


FIGURA 42: Accidentes, infracciones y roturas

La Figura 43 muestra el comportamiento de la velocidad de los vehículos en las tres simulaciones. La velocidad de los vehículos se ve condicionada por varios factores como son la temperatura, la lluvia, y manera general el estado ambiental. En la simulación B este valor fue más alto debido a que los conductores estaban expuestos a una temperatura de 30 grados y además llovía, lo que provoca un aumento de la velocidad pues cada vehículo intenta arribar lo más pronto posible. En las simulaciones A y C las temperaturas eran bastante similares, la diferencia radica en que en la simulación C había lluvia y por ello la velocidad promedio de los vehículos es mayor que la velocidad promedio de A.

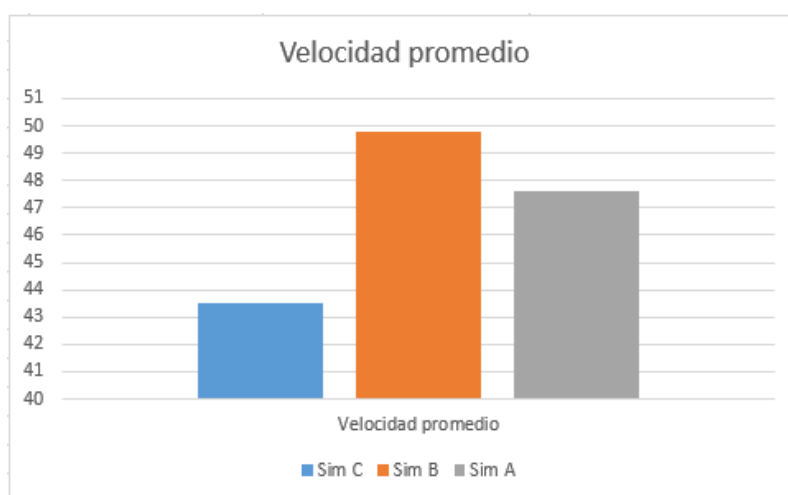


FIGURA 43: Velocidad promedio de los vehículos

El tiempo promedio de espera de los vehículos ante las luces del semáforo es una de las variables más importantes a analizar para decir si las señales colocadas en la simulación han tenido o no un buen impacto. La Figura 44 muestra como en la simulación A los vehículos tuvieron tiempo de espera 0, pues era baja la cantidad de vehículos en la vía. Por otra parte las simulaciones B y

C presentan tiempos de espera más alto, que responden a un mayor número de vehículos en el entorno de simulación y a mayores tiempos de luz roja en el semáforo.

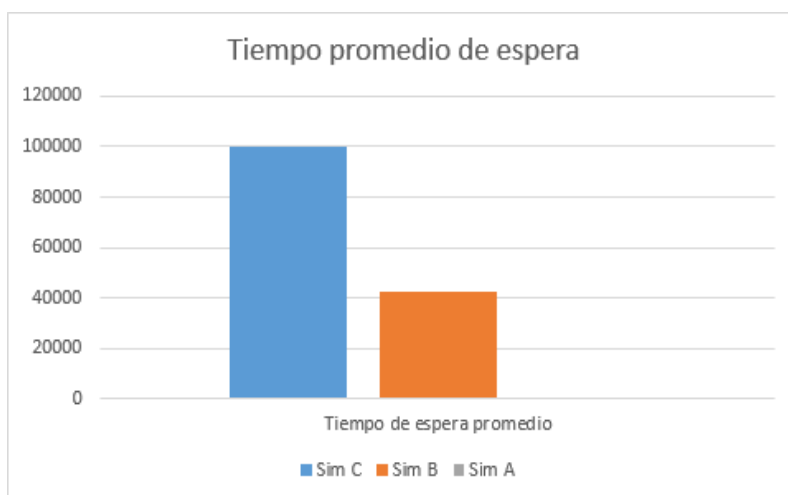


FIGURA 44: Tiempo promedio en espera

3.3. Diseño de Experimentos

Como parte de la validación de la solución propuesta se lleva a cabo un diseño de experimentos (DoE, *Design of experiments*). Este es definido como: “una secuencia de pasos tomados de antemano para asegurar que los datos apropiados se obtendrán de modo que permitan un análisis objetivo que conduzcan a deducciones válidas con respecto al problema planteado” [60]. El DoE desarrollado está orientado al caso: “Ciudad Deportiva semaforizada”. Para ello se siguen las directrices del DoE.

3.3.1. Fase de planeación

Esta primera fase comienza con el **reconocimiento y formulación del problema**. Se quiere realizar una simulación en la Ciudad Deportiva, de forma tal que los vehículos circulen por esta área, incluida la rotonda, teniendo que esperar el menor tiempo posible.

El **rendimiento** es tiempo de espera promedio, en minutos, de los vehículos luego de una simulación de 12 horas. Este tiempo se ve afectado en cada vehículo cuando, está esperando por la luz del semáforo, obstaculizado por un accidente o en cola, es decir, por otro vehículo que a su vez está a la espera. El objetivo es minimizar este tiempo.

Los **factores** identificados se dividen en dos grupos, controlables y no controlables. El primer grupo, acompañado de los **niveles**, está descrito en la Tabla 28. También fueron identificados otros factores que no pueden ser controlados (segundo grupo), pero no por esto menos

importantes, ya que tienen una influencia considerable en el rendimiento. Estos son, cantidad de accidentes, número de infracciones, número de infractores y cantidad de vehículos que llegan a su fin.

TABLA 28: Factores controlables, niveles y unidades de medidas

ID	Factor	Nivel		Unidad
		Bajo (-1)	Alto (1)	
A	Presencia de semáforos en la rotonda	No	Sí	-
B	Velocidad promedio de los vehículos	40	60	Km/h
C	Cantidad de vehículos que arriban	1	3	u/s

3.3.2. Fase de diseño

En esta fase se elige la **resolución de diseño** a aplicar. La cantidad de factores controlables son tres, esto implica que la cantidad total de tratamientos es ocho según la siguiente ecuación:

$$cantidad_de_tratamientos = cantidad_de_niveles^{cantidad_de_factores} = 2^3 = 8 \quad (3.1)$$

Esta cantidad es aceptable (considerablemente pequeña en cuanto a ejecuciones), por lo que se decide aplicar una resolución **factorial completo**. El beneficio más importante que se obtiene con esta resolución de diseño es que se evita la **confusión o alias**, la cual estaría presente de realizar menos ejecuciones que cantidad de tratamientos total [60].

3.3.3. Fase de conducción

Para llevar a cabo la fase de conducción se realizan las 8 ejecuciones con los tratamientos correspondientes. Como herramienta de apoyo se hace uso del software estadístico Minitab [61, 62]. La Tabla 29, porta los resultados en cuanto a rendimiento de cada experimento.

TABLA 29: Rendimiento de cada tratamiento

No.	A: Semáforo	B: Velocidad	C: Arribo	Tiempo
1	1	1	1	31
2	-1	-1	1	12
3	1	-1	-1	28
4	1	-1	1	37
5	-1	1	-1	9
6	1	1	-1	22
7	-1	-1	-1	10
8	-1	1	1	13

3.3.4. Fase de análisis

Minitab, luego de insertado los datos, realiza un procesamiento de estos, que tiene como resultado un conjunto de gráficas que harán posible el análisis final. La gráfica de la Figura 45, ofrece los factores principales influyentes para el rendimiento del problema.

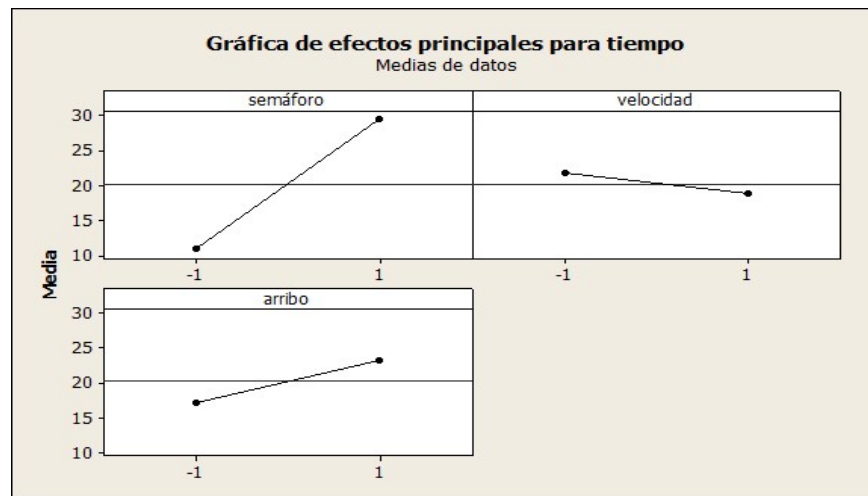


FIGURA 45: Gráfica de efectos principales

En ella se observa que la pendiente del factor A es considerablemente mayor que las pendientes de los factores B y C. Esto sugiere que el efecto principal más influyente es el del factor A (Presencia de semáforos en la rotonda).

Por su parte, la gráfica de Pareto de la Figura 46, muestra una comparativa clara entre el efecto de cada factor o interacción. En esta se confirma que el factor A (Presencia de semáforos en la rotonda) es el influyente, y agrega que no existe ninguna interacción de factores con una influencia considerable.

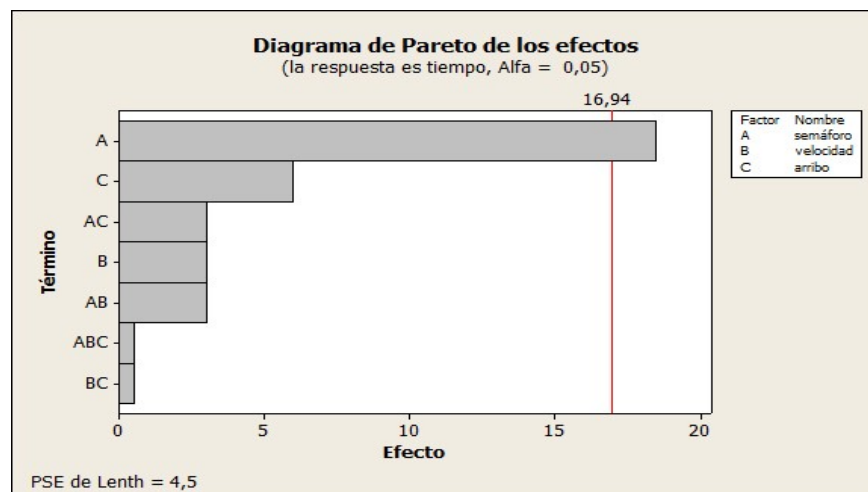


FIGURA 46: Gráfica de Pareto de efecto y rendimiento

Para tener mayor certeza que el factor A (Presencia de semáforos en la rotonda) es el único influyente, Minitab ofrece la gráfica ilustrada en la Figura 47. En esta gráfica se observa que, luego de plotear los puntos de cada efecto, el punto del factor A es el único que dista considerablemente de la recta plotuada.

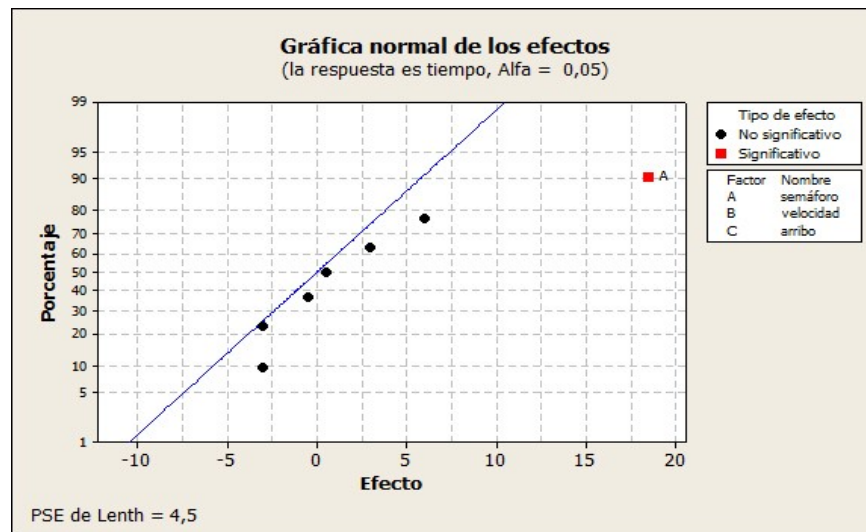


FIGURA 47: Gráfica de probabilidad normal

A la necesidad de conocer cuál es la configuración óptima para este DoE, Minitab genera una gráfica de cubo. La Figura 48, denota a cada vértice del cubo con el resultado del rendimiento. El objetivo del diseño es minimizar el rendimiento, por lo tanto, al buscar el mínimo, se puede apreciar que el menor tiempo de espera promedio de los vehículos es 9 minutos. Entonces, las coordenadas de este vértice indican el tratamiento que logró un mejor rendimiento.

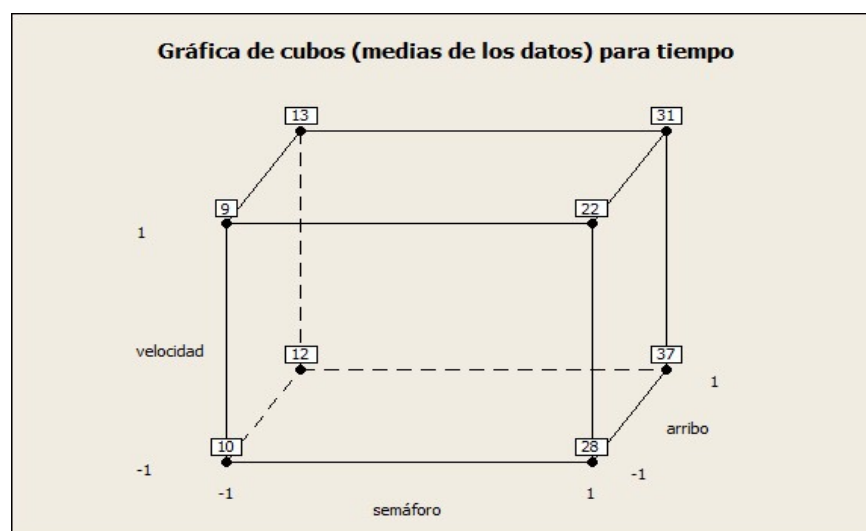


FIGURA 48: Gráfica de cubos

3.3.5. Conclusiones del DoE

Luego de la aplicación del DoE se puede concluir que:

- La elección de un diseño factorial completo evita la confusión o alias.
- Comparado con el arribo de los vehículos y su velocidad, la presencia de semáforos en la rotonda de la ciudad deportiva es el factor más influyente.
- Para el DoE de la Ciudad Deportiva, no existe ninguna interacción de factores con un efecto influyente.
- Todos los tratamientos en los que había presencia de semáforos en la rotonda de la Ciudad Deportiva resultaron generar un tiempo de espera en los vehículos considerablemente mayor a los que no existía semáforo.
- La configuración que logra un mejor rendimiento se muestra en la Tabla 30, con un tiempo de espera promedio de 9 minutos.

TABLA 30: Configuración Recomendada

ID	Factores	Niveles	Valor	Unidad
A	Presencia de semáforos en la rotonda	Bajo (-1)	no	-
B	Velocidad promedio de los vehículos	Alto (1)	60	Km/h
C	Cantidad de vehículos que arriban	Bajo (-1)	1	u/s

3.4. Conclusiones parciales

Luego de validar la herramienta desarrollada se llega a las siguientes conclusiones:

- Para la validación de la herramienta, se seleccionaron pruebas funcionales utilizando la técnica guiada por casos de uso con el objetivo de detectar fallas en el cumplimiento de los requisitos.
- Para la mayoría de los escenarios, los resultados obtenidos se correspondieron con los esperados o bien eran resultados aceptados.
- Con la existencia de más de 100 vehículos al mismo tiempo en la herramienta, se generan excepciones de acceso concurrente.
- Se demostró que la herramienta cumple con sus funcionalidades para la cual fue desarrollada.

Conclusiones generales

En el trabajo se cumplieron los objetivos propuestos, y constituye para los analistas e ingenieros del tránsito una herramienta de simulación para contribuir al proceso de configuración de controladores del tránsito en redes viales. Podemos concluir que:

- La Simulación Basada en Agentes es un método eficaz para describir sistemas complejos como el del transporte con los componentes principales.
- Las malas configuraciones semafóricas, pueden afectar en gran medida al medio ambiente y contribuir a los accidentes.
- Los artefactos de Ingeniería de Software construidos sirvieron para definir el modelado del negocio a automatizar.
- Se modificó la herramienta existente y se añadió la funcionalidad de la luz amarilla para los semáforos.
- La solución fue probada mediante la ejecución de los casos de prueba, con los que se tuvo constancia de la implementación adecuada en la herramienta.
- Se compararon los resultados obtenidos con diferentes juegos de datos demostrando la influencia que tienen los factores añadidos a la simulación en el comportamiento de los vehículos.

Recomendaciones

- Agregar a la simulación otros tipos de semáforos como, por ejemplo, los semáforos de peatones.
- Mejorar la representación gráfica de los vehículos.
- Desarrollar un mecanismo de análisis de datos estadísticos que sugiera configuraciones de buen rendimiento vial basándose en los datos de accidentalidad obtenidos de cada simulación.
- Resolver el problema de acceso concurrente cuando la aplicación está bajo estrés.
- Desarrollar nuevos casos de pruebas en un laboratorio de calidad que abarquen el 100 % de cobertura.

Referencias bibliográficas

- [1] Hillier F. S. *Investigación de operaciones*. McGraw-Hill Interamericana de España, 2010.
- [2] Kelton WD. Law AM, Kelton WD. *Simulation modeling and analysis*. McGraw-Hill New York, 1991.
- [3] M. Curiel. Tipos de simulación. *Curiel*, 2003.
- [4] Michael Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley & Sons, 2009.
- [5] Cubadebate. A partir del viernes 9 funcionarán semáforos en la rotonda de la ciudad deportiva [internet], [citado 30 de abril de 2018]. Disponible en: <http://www.cubadebate.cu/noticias/2015/10/08/a-partir-del-viernes-9-funcionaran-semaforos-en-la-rotonda-de-la-ciudad-deportiva/>, 2015.
- [6] Cub. Rotondas, semáforos y comunicación por descongestionar [internet], [citado 30 de abril de 2018]. Disponible en: <http://www.cubadebate.cu/opinion/2015/12/23/rotondas-semaforos-y-comunicacion-por-descongestionar/>, 2015.
- [7] Dayan Bravo Fraga. *Herramienta de Simulación Basada en Agentes para la evaluación de configuraciones semafóricas en redes viales*. PhD thesis, Universidad Tecnológica de La Habana, José Antonio Echeverría, CUJAE, 2018.
- [8] Comisión económica para Europa. División de Transporte. Convención sobre la señalización vial. *Naciones Unidas*, 2007.
- [9] Alejandro Machado de Leon. *Código de Seguridad Vial*. ONBC, 2016.
- [10] Eastleigh Borough Council. Speed limits and reduction: Speed humps. *Eastleigh Borough Council*, 2006.
- [11] Ministerio de Salud Pública. Dirección de registros médicos y estadísticas de la salud. Anuario estadístico de salud. *Anuario Estadístico de Salud*, 2016.
- [12] ONEI. Estadísticas de oficina nacional de estadísticas e información. cuba [internet] [citado 23 de septiembre de 2020]. Disponible en: <http://www.onei.cu/>, 2020.

- [13] Real Academia Española. *Diccionario de la Lengua Española*. Real Academia Española, 2014.
- [14] Redacción Nacional. ¿cómo el país se propone impedir el deterioro de las vías? *Periódico Granma*, 08 2019.
- [15] Ian Thompson; Alberto Bull. La congestión del tránsito urbano: causas y consecuencias económicas y sociales. *Recursos Naturales e Infraestructura*, 2001.
- [16] Cesvi Colombia. Conducir bajo la lluvia aumenta el riesgo en las vías. *Auto Crash*, 48, 2018.
- [17] A. de la Casa; A. C. Ravelo. Assessing temperature and humidity conditions fot dairy cattle in cordoba, argentina. *Int. J. Biometeorol*, 2003.
- [18] Stuart Newstead. An investigation into the relationship between vehicle colour and crash risk. *Monash University. Accident Research Center*, 2007.
- [19] *FIPA, Foundation for Intelligent Physical Agents*.
- [20] Mailyn Moreno Espino. Metodologías orientadas a agentes: un estudio comparativo. Master's thesis, Instituto Superior Politécnico: José Antonio Echeverría, 2006.
- [21] Zead A. A uml profile for agent-based development. *Metainformatics*, 2003.
- [22] Abar S.; Kinoshita T. A knowledge-based strategy for the automated support to network management tasks. *IEICE Trans*, 2010.
- [23] Moon Y. B. *Simulation modelling for sustainability: a review of the literature*. Int J Sustain Eng., 2017.
- [24] Lemarinier P. Abar S, Theodoropoulos GK. *Agent Based Modelling and Simulation tools: A review of the state-of-art software*. Comput Sci Rev., 2017.
- [25] Paula Nicole Roldán. Modelo matemático. *Economipedia*, 2020.
- [26] IA Colectivo de Profesores. Conferencia: Incertidumbre difusa, 2018.
- [27] Stuart J. Russell; Peter Norvig. *Artificial Intelligence. A Modern Approach*. PRENTICE HALL, 2010.
- [28] Carlos Gonzalez Morcillo. Lógica difusa. una introducción práctica. *Técnicas de Softcomputing*, 2017.
- [29] Capote Fernández JL. Delgado Fernández T. Semántica espacial y descubrimiento de conocimiento para desarrollo sostenible. *Marco Teórico Infraestruct Datos Espaciales*, 2009.
- [30] Kemp K. Encyclopedia of geographic information science. *SAGE*, 2008.

- [31] SQLite. Sqlite home page [internet]. [citado 10 de mayo de 2018]. Disponible en: <https://www.sqlite.org/index.html>, 2015.
- [32] OpenStreetMap. Spatialite-tools: Osm tools [internet]. [citado 10 de mayo de 2018]. Disponible en: <https://www.gaia-gis.it/fossil/spatialite-tools/wiki?name=OSM+tools>, 2015.
- [33] Mancebo Q. S. Aprendiendo a manejar los sig en la gestión ambiental. *Libr SIG Esp.*, 2008.
- [34] Beinat E. Kothuri R, Godfrind A. *Pro oracle spatial for oracle database 11g*. Dreamtech Press, 2008.
- [35] Sánchez Ansola E. *Servicios basados en localización para una infraestructura de datos espaciales*. PhD thesis, Instituto Superior Politécnico: José Antonio Echeverría, 2015.
- [36] OpenStreetMap. Openstreetmap [internet]. [citado 15 de mayo de 2018]. Disponible en: <https://www.openstreetmap.org/>, 2018.
- [37] Sergio Ariel König; Leandro Matías Leuci. Configuración eficiente de ciclos semafóricos de la ciudad de córdoba. enfoque sistémico, 2017.
- [38] Jaganath Pablo Emmanuel León Carrascoza. Diseño e implementación de sistema de conmutación remoto para semáforos, 2019.
- [39] Junchen Jina; Xiaoliang Maa; Iisakki Kosonenb. An intelligent control system for traffic lights with simulation-based evaluation. *Control Engineering Practice*, 2017.
- [40] OSCAR MANUEL ALMONACID MANSILLA. Simulación digital de tráfico para intersecciones señalizadas por semáforo, bajo ambiente tridimensional, 2007.
- [41] Edgar Gómez Hernández. Desarrollo de un modelo de simulación vehicular para la mejora en la sincronización de semáforos. Master's thesis, Instituto Nacional de Astrofísica, Óptica y Electrónica, 2009.
- [42] Ashfaq H. Farooqi; Ali Munir; A. Rauf Baig. The: traffic light simulator and optimization using genetic algorithm. *International Conference on Computer Engineering and Applications*, 2011.
- [43] Khodakaram Salimifard; Mehdi Ansari. Modeling and simulation of urban traffic signals. *International Journal of Modeling and Optimization*, 2013.
- [44] José A. Castán; Salvador Ibarra; Julio Laria; Javier Guzmán; Emilio Castán. Control de tráfico basado en agentes inteligentes. *Polibitis*, 2014.
- [45] Orlando Fonseca Guilarte; Gabriel S. Pujol Fariña; Sira M. Allende Alonso; Gemayqzel Bouza Allende. Herramienta computacional para sugerir y validar políticas de luces de semáforos. *13er Simposio Argentino de Investigación Operativa*, 2015.

- [46] Orlando Fonseca Guilarte; Gabriel S. Pujol Fariña; Sira M. Allende Alonso; Gemayqzel Bouza Allende. Una herramienta para simular y analizar el flujo de tráfico en un cruce regulado por semáforos con tiempos de ciclos variables. *Revista Cubana de Ciencias Informáticas*, 2014.
- [47] Carlos Dómine Rueda. Simulación de tráfico en circuitos urbanos para analizar el efecto de diferentes tipos de conducción, 2015.
- [48] RICARDO OLIVARES GALLEG0; DANIEL BORRAJO MILLÁN. Modelo de simulación y planificación de tráfico urbano mediante semáforos inteligentes y sumo, 2015.
- [49] Ignacio Alejandro Palermo; Julio Renato Rivera Laguna. Herramienta para la edición y evaluación de redes de tráfico vehicular, 2017.
- [50] N. C. Sarkar; A. Bhaskar; Z. Zheng; M. P. Miska. Microscopic modelling of area-based heterogeneous traffic flow: Area selection and vehicle movement. *Transp. Res. Part C Emerg. Technol.*, 111:373–396, Feb 2020.
- [51] A. Wegener; M. Piórkowski; M. Raya; H. Hellbrück; S. Fischer; J.-P. Hubaux. Traci. In *Proceedings of the 11th communications and networking simulation symposium*, number 08, page 155, 2008.
- [52] Booch G. Rumbaugh J, Jacobson I. *Unified modeling language reference manual, the*. Pearson Higher Education, 2004.
- [53] Roger Pressman. *Ingeniería del software. Un enfoque práctico*. The McGraw Hill, 2010.
- [54] Rumbaugh J. Jacobson I, Booch G. *El proceso unificado de desarrollo de software/The unified software development process*. Pearson Educación, 2000.
- [55] Sean Luke. *Multiagent Simulation And the MASON Library*. Department of Computer Science George Mason University, 2015.
- [56] Richard Johnsonbaugh. *MATEMÁTICAS DISCRETAS*. PEARSON EDUCACIÓN, 6 edition, 2005.
- [57] OpenWeatherMap. Openweathermap [citado el 10 de noviembre de 2020]. Disponible en: <http://owm.org/>, 2020.
- [58] E. Gamma.; R. Helm; R. Johnson; J. Vlissides. *Design Patterns, Elements Of Reusable Object Oriented Software*. Read Addison-Wesley, 2000.
- [59] C. Guerrero; J. Suarez; L. Gutierrez. Patrones de diseño gof en el contexto de procesos de desarrollo de aplicaciones orientadas a la web. *Inf. Tecnológica*, 2013.
- [60] M. J. Anderson; P J. Whitcomb. *Design of experiments*. Wiley Online Library, 2000.

- [61] Minitab INC. *MINITAB statistical software. Minitab Release*, 2000.
- [62] Minitab INC. Minitab [internet] [citado 17 de diciembre de 2020]. *Disponible en:* <http://www.minitab.com/es-mx/>, 2018.
- [63] Cioffi-Revilla C. Luke S. *Mason: A multiagent simulation environment*. Simulation, 2005.
- [64] MASON. Mason [internet], [citado 14 de marzo de 2017]. *Disponible en:* <https://cs.gmu.edu/eclab/projects/mason/docs/>, 2017.
- [65] Mark Coletti. *The GeoMason Cookbook*. Department of Computer Science George Mason University, 2013.
- [66] JadeSite. Java agent development framework [internet], [citado 27 de junio de 2017]. *Disponible en:* <http://jade.tilab.com/>, 2017.
- [67] Jordan Ramos Piloto. Simulación basada en agentes para el chequeo de requisitos tempranos en i*. Master's thesis, Universidad Tecnológica de La Habana, José Antonio Echeverría, CUJAE, 2015.
- [68] Pokahr A. *Jadex: A BDI reasoning engine. Multi-Agent Program*. H, 2005.
- [69] Pokahr A. *Jadex user guide*. Hambg Ger, 2007.
- [70] NetLogo. Netlogo home page [internet]. [citado 26 de junio de 2017]. *Disponible en:* <http://ccl.northwestern.edu/netlogo/>, 2017.
- [71] Collier N. *Repast: An extensible framework for agent simulation*. Univ Chicago Soc Sci Res, 2003.
- [72] freelancerviet.vn. Gama is a modeling and simulation development environment for building spatially explicit agent-based simulations. *Disponible en:* <http://gama-platform.org/references>, 2017.

ANEXO A

Plataformas de Agentes

A.1. MASON

MASON (*Stands for Multi-Agent Simulator Of Neighborhoods... or Networks... or something...*) es una plataforma de simulación basada en agente, de eventos rápidos y discretos, diseñado para ser la base para grandes simulaciones en Java de propósito personalizado, y también para proporcionar funcionalidad más que suficiente para muchas necesidades de simulación de peso ligero. Contiene una biblioteca de modelos y un conjunto opcional de herramientas de visualización en 2D y 3D [63, 64].

Esta plataforma constituye un esfuerzo conjunto entre el Laboratorio de Computación Evolutiva de la Universidad George Mason y el Centro de Complejidad Social. Sus diseñadores son: Sean Luke, Gabriel Catalin Balan, Keith Sullivan y Liviu Panait, con ayuda de Claudio Cioffi Revilla, Sean Paus, Keith Sullivan, Daniel Kuebrich, Joey Harrison y Ankur Desai [63].

Características:

- 100 % Java (1.3 o superior).
- Rápido, portátil y bastante pequeño.
- Los modelos son completamente independientes de la visualización, que se pueden agregar, quitar o cambiar en cualquier momento.
- Los modelos pueden ser controlados y recuperados, y migrados dinámicamente a través de las plataformas.
- Puede producir resultados idénticos entre plataformas.
- Los modelos son independientes y pueden ejecutarse dentro de otros marcos de trabajo y aplicaciones Java.

- Visualización 2D y 3D.
- Puede generar capturas de pantalla, películas *QuickTime*, gráficos.

Objetivos de diseño:

- Posibilidad de ejecutar muchos tipos de simulaciones
- Sistemas sociales complejos
- Modelado físico
- Inteligencia artificial
- Eficiente con grandes números de agentes
- Alto grado de modularidad y flexibilidad
- Separar las herramientas de visualización
- *Check-pointing* y recuperación

Las bibliotecas se proporcionan para visualizar en 2D y en 3D (utilizando Java3D), manipular el modelo gráficamente, tomar capturas de pantalla y generar películas (utilizando Java Media Framework) [55].

GeoMason: Como parte de la integración con GIS, MASON cuenta con una extensión opcional llamada GeoMason la cual añade soporte para datos geoespaciales y se publica bajo la Licencia Libre Académica [65].

A.2. JADE

JADE (*Java Agent DEvelopment Framework*) es un marco de trabajo de software totalmente implementado en el lenguaje Java. Simplifica la implementación de sistemas multi-agente a través de un middleware que cumple con las especificaciones FIPA y mediante un conjunto de herramientas gráficas que soportan las fases de depuración e implementación. Un sistema basado en JADE puede distribuirse entre máquinas (que ni siquiera necesitan compartir el mismo sistema operativo) y la configuración se puede controlar a través de una interfaz gráfica de usuario (GUI, *Graphical User Interface*) remota. La configuración puede cambiarse incluso en tiempo de ejecución moviendo agentes de una máquina a otra, según sea necesario. JADE está completamente implementado en lenguaje Java y el requisito mínimo del sistema es la versión 5 de JAVA (el entorno de tiempo de ejecución o el JDK) [66].

JADE es software libre y es distribuido por Telecom Italia, titular de los derechos de autor, en código abierto bajo los términos y condiciones de la licencia LGPL (Licencia Pública General Menor Versión 2) [67].

JADE está entre las plataformas más conocidas y utilizadas debido a las funcionalidades que permite desarrollar. Entre las mismas, facilita construir sistemas de agentes para el manejo de información y recursos a través de redes de computadoras, bajo la complacencia de las especificaciones FIPA, para sistemas multi-agente interoperables [67].

JADE es un middleware que consiste en un entorno de ejecución, donde los agentes pueden vivir, y estar activos en determinado host, antes de que otros puedan ser ejecutados en el mismo. Ofrece una biblioteca de clases, que los programadores pueden usar (directamente o modificando las mismas) para desarrollar los agentes, así como una suite de herramientas gráficas, que permite administrar y monitorear la actividad de los agentes que se estén ejecutando [66].

La plataforma puede ser distribuida a través de sistemas heterogéneos, y brinda soporte a agentes móviles. La arquitectura de la comunicación ofrece una flexible y eficiente mensajería, donde JADE crea y administra una cola de mensajes ACL privada para cada agente. Todo el estándar FIPA de comunicaciones ha sido implementado: protocolos de interacción, ACL, lenguaje contenido, esquemas, ontologías y protocolos de transportación. El mecanismo para el transporte de mensajes utiliza Java RMI.

Dicha plataforma está compuesta por contenedores de agentes, que pueden estar distribuidos en una red de computadoras. Los agentes viven en estos contenedores, que son procesos Java que provee el entorno de ejecución JADE, y todos los servicios necesarios para el hosting y ejecución de agentes. Hay un contenedor especial (Principal), que es el primero en ser ejecutado, y entre sus funcionalidades está registrar las referencias de otros contenedores que componen la plataforma, así como todos los agentes, incluyendo el estado actual de los mismos y localización, además del control de dos agentes especiales, uno denominado AMS, encargado de supervisar la plataforma y el DF, quien posee una lista de servicios registrados por defectos que pueden ser utilizados por los agentes.

A.3. JADEx

Distintas plataformas dan soporte al desarrollo de aplicaciones basadas en sistemas multi-agente, identificándose dos grupos: por un lado, aquellas que obedecen las especificaciones FIPA, destacándose JADE, mientras que, en otro, se encuentran las plataformas centradas en el razonamiento, teniendo como base la arquitectura BDI (*Believe, Desire, Intention*). Estas diferencias, constituyen la principal motivación de la plataforma Jadex BDI, la cual cubre las particularidades de ambos grupos, en un único entorno, cumpliendo con los estándares FIPA, así como dando soporte al razonamiento BDI [68].

Este proyecto nace en el 2002 como parte de investigaciones entre el Departamento de ciencias de la computación de la Universidad de Hamburgo en Alemania y el MedPage. JADEX es un marco de trabajo basado en JADE, para ingeniería de razonamiento, bajo especificaciones FIPA, que facilita el desarrollo de agentes a través de los lenguajes de programación Java y de marcado XML. Es distribuido bajo licencia GNU- LGPL [69].

En esta plataforma, no se introduce un nuevo lenguaje de programación, lo que deviene en una gran ventaja, dado que los agentes pueden ser implementados bajo los conocimientos del estado del arte, así como los principales entornos de desarrollo integrado como Eclipse e IntelliJ IDEA. Otra ventaja, radica en que los agentes representan componentes activos con capacidades de razonamiento individual, lo que se traduce en el hecho de que pueden exhibir comportamientos reactivos y proactivos [69].

JADEX incorpora el modelo BDI en un agente JADE mediante creencias, metas y planes. Las creencias pueden ser cualquier tipo de objeto Java y ser almacenadas en una base de creencias. Las metas representan motivaciones concretas (por ejemplo, estados a lograr) que tienen influencia en el comportamiento del agente y, para lograrlas, se ejecutan planes implementados como clases Java. Todo este estado de los agentes, es capturado en un Archivo de Definición de Agente (ADF). El ADF es un fichero XML donde los desarrolladores definen creencias iniciales y metas mediante expresiones establecidas para tales propósitos. El lenguaje para estas expresiones es Java, con una extensión de constructores OQL.

A.4. NetLogo

NetLogo es un entorno de modelado programable multi-agente para simular fenómenos naturales y sociales. Es utilizado por decenas de miles de estudiantes, profesores e investigadores de todo el mundo. Fue escrito por Uri Wilensky en 1999 y ha estado en continuo desarrollo desde entonces [70].

Es particularmente adecuado para modelar sistemas complejos que se desarrollan con el tiempo. Los modeladores pueden dar instrucciones a cientos o miles de “agentes” que funcionan de forma independiente. Esto hace posible explorar la conexión entre el comportamiento micro-nivel de los individuos y los patrones macro-nivel que emergen de su interacción [70].

NetLogo permite a los estudiantes abrir simulaciones y “jugar” con ellas, explorando su comportamiento bajo diversas condiciones. También es un entorno de creación que permite a los estudiantes, profesores y desarrolladores de planes de estudio crear sus propios modelos. Es bastante simple para estudiantes y profesores, pero lo suficientemente avanzado como para ser una herramienta poderosa para los investigadores en muchos campos [70].

Tiene una amplia documentación y tutoriales. También viene con la Biblioteca de Modelos, una gran colección de simulaciones pre-escritas que se pueden utilizar y modificar. Estas simulaciones abordan áreas de contenido en las ciencias naturales y sociales, incluyendo biología y medicina, física y química, matemáticas e informática, y economía y psicología social. Existen varios planes de estudio basados en modelos que utilizan NetLogo y otros están en desarrollo [70].

NetLogo es la próxima generación de la serie de lenguajes de modelado multi-agente, incluyendo StarLogo y StarLogoT. NetLogo se ejecuta en la máquina virtual Java, por lo que funciona en todas las plataformas principales (Mac, Windows, Linux, et al). Se ejecuta como una aplicación de escritorio. También se admite la operación en línea de comandos [70].

Características:

■ **Sistema:**

- Libre, código abierto
- Portabilidad: se ejecuta en Mac, Windows, Linux y otros
- Soporte de conjunto de caracteres internacional

■ **Programación:**

- Totalmente programable
- Sintaxis accesible
- Los agentes móviles (tortugas) se mueven sobre una cuadrícula de agentes estacionarios (parches)
- Los agentes de enlace conectan a las tortugas para hacer redes, gráficos y agregados
- Amplio vocabulario de primitivas de lenguaje incorporadas
- Valores de funciones de primera clase (también conocidos como procedimientos anónimos o cierres)
- Las ejecuciones son reproducibles en multiplataforma

■ **Ambiente:**

- Centro de mando para la interacción directa
- Interfaz constructor, botones, controles deslizantes, interruptores, selectores, monitores, cuadros de texto, notas, área de salida
- Pestaña de información para anotar su modelo con texto e imágenes con formato
- HubNet: simulaciones participativas utilizando dispositivos en red
- Monitores de agente para inspeccionar y controlar agentes

- Funciones de exportación e importación (datos de exportación, guardar y restaurar el estado del modelo, hacer una película)
- *BehaviorSpace*, una herramienta de código abierto utilizada para recopilar datos de múltiples ejecuciones paralelas de un modelo
- *System Dynamics Modeler*
- NetLogo 3D para modelar mundos 3D
- El modo *Headless* permite realizar lotes desde la línea de comandos

■ **Visualización:**

- Parcelas de líneas, barras y dispersiones
- El control deslizante de velocidad le permite avanzar rápidamente su modelo o verlo en cámara lenta
- Ver su modelo en 2D o 3D
- Formas vectoriales escalables y giratorias

A.5. RePast

Repast es un conjunto de herramientas de simulación y modelado basado en agentes, libre y de código abierto. Actualmente están disponibles tres plataformas Repast, cada una de las cuales tiene las mismas características principales pero un entorno diferente para estas características. Repast Symphony (Repast S) extiende la cartera de Repast ofreciendo un nuevo enfoque para el desarrollo y la ejecución de la simulación.

RePast es un marco de trabajo para la simulación basada en agentes creado por *Social Science Research Computing* de la Universidad de Chicago. Proporciona una biblioteca integrada de clases para crear, ejecutar, mostrar y recopilar datos de una simulación basada en agentes [71].

El modelo RePast típico contiene un conjunto de agentes. Estos agentes pueden o no ser homogéneos, o tal vez estos agentes están dispuestos en una jerarquía (una empresa y sus empleados, por ejemplo). Independientemente de su composición, cada agente tiene cierto comportamiento, las interacciones de las cuales un modelador está interesado en explorar.

El tiempo de ejecución de Repast S está diseñado para incluir funciones avanzadas de almacenamiento, visualización y activación de comportamiento del agente, así como nuevas instalaciones para el análisis y la presentación de datos. Este artículo presenta la arquitectura y las características principales del sistema de tiempo de funcionamiento Repast S y discute cómo se ajusta Repast S dentro de la cartera más grande de Repast [71].

A.6. GAMA

GAMA es un entorno de desarrollo de simulación y modelado para construir simulaciones basadas en agentes.

Entre sus características principales se encuentran:

- Múltiples dominios de aplicación: Garantiza la posibilidad de utilizar GAMA para cualquier dominio de aplicación que desee.
- Lenguaje de alto nivel e intuitivo basado en agentes: Permite escribir sus modelos fácilmente usando GAML como lenguaje.
- GIS y modelos dirigidos por datos: Ofrece instanciar agentes de cualquier conjunto de datos, incluyendo datos GIS, y ejecutar simulaciones a gran escala (hasta millones de agentes).
- Interfaz de usuario declarativa: Se pueden declarar interfaces que soporten inspecciones profundas en agentes, paneles de acción controlados por el usuario, pantallas multidimensionales 2D / 3D y aspectos de agente.

Consiste en una sola aplicación que se basa en la arquitectura RCP proporcionada por Eclipse. Dentro de este único software de aplicación, a menudo denominado plataforma, los usuarios pueden realizar, sin necesidad de software adicional de terceros, la mayoría de las actividades relacionadas con el modelado y la simulación, es decir, edición de modelos y simulación, visualización y exploración utilizando herramientas dedicadas [72].

Se logra, en primer lugar, proporcionar algunos antecedentes sobre las nociones importantes que se encuentran en toda la plataforma, especialmente las del espacio de trabajo y los proyectos y explicar cómo organizar y navegar a través de los modelos. Luego una vista a la edición de modelos y sus diversas herramientas y componentes (editores dedicados y herramientas relacionadas, por supuesto, pero también validadores). Finalmente, se muestra cómo ejecutar experimentos en estos modelos y el soporte que la interfaz de usuario puede proporcionar a los usuarios en esta tarea [72].

En Resumen, GAMA es una plataforma de desarrollo novedosa que provee de recursos amplios para llevar a cabo simulaciones sociales basada en agentes, incluye integración nativa con archivos de datos espaciales para su posterior interacción con agentes sociales. No presenta mecanismos de importar archivos tropos que utiliza el lenguaje de modelado i*. Es una tecnología que cuenta con un respaldo amplio de documentación en cuanto a guías, tutoriales, fórum, etc. Su implementación se rige por un lenguaje de programación propio, relativamente similar a Java. Cuenta con el IDE eclipse gama-plataform, el cual brinda métodos de detección de errores

de código, ejemplos, etc. Sin duda, esta constituye una propuesta conveniente como base de la arquitectura para el desarrollo de la herramienta en cuestión.