



**UNIVERSIDAD
NACIONAL DE LOJA**

Área de la Energía las Industrias y los Recursos Naturales



No Renovables

Carrera de Ingeniería en Sistemas

“Desarrollo y simulación de un sistema multiagente para la comunicación de semáforos para encontrar la ruta optima mediante grafos”

Tesis previa a la Obtención del título de Ingeniero en Sistemas

Autores:

- Loarte Caraguay Robert Iván.
- Quizhpe Vásquez Bolívar Rolando.

Director: Ing. Paz Arias Henry Patricio, Mg.Sc.

Loja-Ecuador

2015

CERTIFICACIÓN DEL DIRECTOR

Mg. Sc. Henry Patricio Paz Arias.

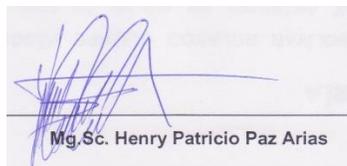
DOCENTE DE LA CARRERA DE INGENIERÍA EN SISTEMAS, DEL ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES DE LA UNIVERSIDAD NACIONAL DE LOJA.

CERTIFICA:

Haber asesorado y revisado detenida y minuciosamente durante todo su desarrollo, el Proyecto de Fin de Carrera titulado: “**DESARROLLO Y SIMULACIÓN DE UN SISTEMA MULTIAGENTE PARA LA COMUNICACIÓN DE SEMÁFOROS PARA ENCONTRAR LA RUTA ÓPTIMA MENDIANTE GRAFOS**” . Realizado por los postulantes Bolívar Rolando Quizhpe Vásquez y Robert Iván Loarte Caraguay.

Por lo tanto, autorizo proseguir los trámites legales pertinentes para su presentación y defensa.

Martes 10 de Febrero de 2015.



Mg.Sc. Henry Patricio Paz Arias

Mg.Sc. Henry Patricio Paz Arias.

DIRECTOR DEL PFC

AUTORÍA

Nosotros: **BOLÍVAR ROLANDO QUIZHPE VÁSQUEZ y ROBERT IVÁN LOARTE CARAGUAY** declaramos ser autores del presente trabajo de tesis y eximimos expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente aceptamos y autorizamos a la Universidad Nacional de Loja, la publicación de nuestra tesis en el Repositorio Institucional – Biblioteca Virtual.



Firma:

Cédula: 1104607120

Fecha: 20 de Mayo de 2015



Firma:

Cédula: 1104149578

Fecha: 20 de Mayo de 2015

CARTA DE AUTORIZACIÓN

Nosotros: **BOLÍVAR ROLANDO QUIZHPE VÁSQUEZ y ROBERT IVÁN LOARTE CARAGUAY**, declaramos ser autores de la tesis titulada: “**DESARROLLO Y SIMULACIÓN DE UN SISTEMA MULTIAGENTE PARA LA COMUNICACIÓN DE SEMÁFOROS PARA ENCONTRAR LA RUTA ÓPTIMA MEDIANTE GRAFOS**”, como requisito para optar al grado de : **INGENIERO EN SISTEMAS**; AUTORIZO AL Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos , muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Intelectual:

Los Usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a veinte días del mes de Mayo del dos mil quince.



Firma:

Autor: Bolívar Rolando Quizhpe Vásquez.

Cédula: 1104607120

Dirección: Cdla. 8 de Diciembre.

Correo Electrónico: boloqv.88@gmail.com

Teléfono: 072562617 **Celular:** 0986756385



Firma:

Autor: Robert Iván Loarte Caraguay.

Cédula: 1104149578

Dirección: Barrio Belén.

Correo Electrónico: robert.loarte@gmail.com

Teléfono: 072552191 **Celular:** 0967034717

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Henry Patricio Paz Arias, Mg. Sc.

Tribunal de Grado: Ing. Luis Roberto Jácome Galarza, Mg. Sc.

Ing. Marco Augusto Ocampo Carpio, Mg. Sc.

Ing. Waldemar Victorino Espinoza Tituana, Mg. Sc.

AGRADECIMIENTO

Expresamos un agradecimiento a la Universidad Nacional de Loja por todos estos años de formación académica

De manera especial a nuestro director de la presente tesis y a todos los docentes que nos ayudaron en la formación académica a lo largo de estos años.

LOS AUTORES

DEDICATORIA

Dedico el presente trabajo primeramente a Dios por permitirme llegar a este momento tan especial de mi vida, ya que me ha dado fortaleza para continuar cuando he estado a punto de caer y me ha enseñado a valorarlo cada día más. A mi madre por estar siempre a mi lado durante mi trayectoria estudiantil y de vida enseñándome a no desfallecer ni rendirme ante nada y siempre perseverar a través de sus sabios concejos. A mi padre sé que está orgulloso de la persona en la cual me he convertido. A mis hermanos Cesar, Sandra, Gladys y Karina quienes con sus consejos me han sabido ayudar a superar los malos momentos y han estado siempre conmigo. A mi prima Paulina quien está pendiente de mis logros y ha sido una hermana más que me ayuda sin recompensa alguna. A mis amigos Fernando, Miguel y Robert gracias por su apoyo sin ustedes no existiera esta tesis ahora y finalmente a una persona q la llevo en mi mente y mi corazón.

Bolívar Rolando Quizhpe Vásquez

A Todos los que me conocen, amigos y enemigos.

Robert Loarte

CESIÓN DE DERECHOS

Robert Iván Loarte Caraguay y Bolívar Rolando Quizhpe Vásquez autores principales del presente trabajo de titulación autoriza a la Universidad Nacional de Loja, al Área de la Energía, las Industrias y los Recursos Naturales No Renovables y por ende a la Carrera de Ingeniería en Sistemas hacer uso del mismo en lo que se estime sea conveniente.

a. Título

“DESARROLLO Y SIMULACIÓN DE UN SISTEMA MULTIAGENTE PARA LA COMUNICACIÓN DE SEMÁFOROS PARA ENCONTRAR LA RUTA ÓPTIMA MEDIANTE GRAFOS”

b. Resumen

En la actualidad se ha adquirido un gran interés en el desarrollo de tecnología para la implementación de ciudades inteligentes y dar soluciones a problemas que se viven día a día, uno de esos problemas es el tráfico vehicular que existen en las ciudades, si bien es cierto existen semáforos para controlar el tráfico, estos son incapaces de tomar decisiones por sí mismos en caso de existir aglomeraciones o accidentes, produciendo grandes desventajas tanto para los conductores y peatones económicamente, y el impacto hacia el medio ambiente.

Bajo este panorama, el presente trabajo trata sobre el desarrollo y simulación de un sistema multiagente que serán representados por semáforos con el objetivo de que estos se comuniquen a través de un lenguaje establecido y logren tomar decisiones de manera autónoma, esto desarrollado a través de una arquitectura denominada BDI (Beliefs: Creencias, Desires: Deseos, Intentions: Intenciones) y a través de JADE (Framework para el desarrollo de agentes en Java) con FIPA-ACL que soporta la coordinación de múltiples agentes y proporciona un lenguaje estándar de comunicación entre ellos.

Los semáforos se comunican cuando se produce un evento, por ejemplo: Accidente, aglomeración y cuando exista un nuevo agente; se comunican los semáforos más cercanos con el semáforo donde se produjo el evento. Cada vez que exista un accidente o aglomeración, estos semáforos se marcarán en el mapa, dando la posibilidad al usuario de evitar esos semáforos y trazando una ruta hacia su lugar de destino.

Summary

Nowadays there is a new interest in the development of technology for the implementation of intelligent cities and provide solutions to problems on the daily living. One of those problems is the vehicular traffic that exist in cities, although it is true there are traffic lights to control traffic, but they are incapable of making decisions for themselves in cases of any agglomerations or accidents happens, that causes great disadvantages for both drivers and pedestrians economically, and the impact to the environment.

Under these circumstances, the following research paper discuss the development and simulation of a multi-agent system that will be represented by the traffic lights in order that these devices get to communicate through a established language and make decisions autonomously, it has been developed through a architecture called BDI (Beliefs: Beliefs, Desires: Desires, Intentions: Intentions) and through JADE (Framework for the development of agents in Java) with FIPA-ACL that supports the coordination of multiple agents and provides a standard language of communication between them.

Traffic lights communicate when an event occurs, for example: Car accidents, agglomeration and when a new agent exists; nearby traffic lights communicate with the traffic lights where the accident took place. Whenever there is an accident or agglomeration, these lights will be marked on the map, giving the user the possibility to avoid those traffic lights and plotting a route to his/her destination.

Índice de Contenidos

CERTIFICACIÓN DEL DIRECTOR.....	I
AUTORÍA.....	II
CARTA DE AUTORIZACIÓN.....	V
AGRADECIMIENTO	IV
DEDICATORIA	V
CESIÓN DE DERECHOS.....	VI
a. Título	1
b. Resumen.....	2
Summary	3
Índice de Contenidos.....	4
Índice de Figuras	9
Índice de Tablas	12
c. Introducción	13
d. Revisión Literaria	15
1. Capítulo I: Semáforos.....	15
1.1. Tráfico vehicular en la ciudad de Loja.....	15
1.2. Semáforos Inteligentes.	16
1.2.1. Tipos de semáforos inteligentes.	16
1.2.1.1. Semáforos inteligentes con RFID.	16
1.2.1.2. Semáforo Inteligente usando Redes de sensores Inalámbricos.	17
1.2.1.3. Semáforos Inteligentes mediante Procesamiento de Imágenes.	17
1.2.1.4. Semáforo Inteligente basado en Inteligencia Artificial.....	18
2. Capítulo II: Sistemas Multiagentes.....	19

2.1.	Definición de Agentes.....	19
2.2.	Sistemas Multiagente.....	20
2.3.	Comunicación entre agentes.....	20
2.4.	Modelos de comunicación.....	21
2.5.	Protocolos de información.....	22
2.5.1.	FIPA-ACL.....	22
2.6.	Teoría de los actos del habla.....	23
2.7.	Arquitectura de Agentes Inteligentes.....	24
2.7.1.	Arquitectura deliberativa.....	24
2.7.2.	Arquitecturas reactivas.....	26
2.7.3.	Arquitectura Híbrida.....	26
2.8.	Metodologías para el desarrollo de agentes inteligentes.....	26
2.9.	GAIA.....	27
2.9.1.	Etapa de análisis.....	27
2.9.1.1.	El Modelo de Roles.....	28
2.9.1.2.	El Modelo de Interacción.....	28
2.9.1.3.	El modelo de agentes.....	28
2.9.1.4.	El modelo de servicios.....	28
2.9.1.5.	El modelo de conocimiento o de relaciones.....	29
2.10.	JADE.....	29
2.10.1.	Plataforma y Arquitectura de JADE.....	30
3.	Capítulo III: GRAFOS.....	32
3.1.	Introducción.....	32

3.2.	Terminología de grafos [24]:.....	33
3.3.	DEFINICIONES.....	33
3.4.	TERMINOLOGÍA y CONCEPTOS	33
3.4.1.	Grafos dirigidos y no dirigidos.....	33
3.4.2.	Incidencia, adyacencia y grado de un vértice	34
3.4.3.	Grafos simples y multígrafos.....	35
3.4.4.	Camino, bucle y ciclo.....	36
3.5.	Recorrido de un grafo:	36
e.	Materiales y Métodos	38
1.	Materiales	38
2.	Métodos.....	38
a.	Método científico.....	38
b.	Método Analítico.	38
c.	Metodología de desarrollo de agentes Inteligentes (GAIA)	39
f.	Resultados	40
1.	PRIMERA FASE: Desarrollar los agentes inteligentes para la comunicación de semáforos en tiempo real.....	40
1.1.	Estudio de los agentes deliberativos.	40
1.2.	Selección de herramientas a utilizar para el desarrollo del sistema de agentes inteligentes	41
1.3.	Análisis y diseño de los agentes deliberativos.....	43
1.3.1.	Metodología de desarrollo.....	43
1.3.2.	Análisis.	43
1.3.2.1.	Modelo de roles.	43

1.3.2.2.	Modelo de Interacciones.	46
1.3.3.	Diseño del sistema	48
1.3.3.1.	Modelo de agentes.	49
1.3.3.2.	Modelo de comunicaciones.	50
1.4.	Programar los agentes para la comunicación de semáforos	51
1.4.1.	Inicio del contenedor principal.	51
1.5.	Permitir la creación e integración de los nuevos agentes deliberativos.	57
2.	SEGUNDA FASE: Desarrollar algoritmos que permitan establecer rutas óptimas mediante la ubicación de semáforos.	59
2.1.	Analizar que algoritmo se utilizará para la creación de los grafos.	59
2.2.	Diseñar grafos que permitan trazar la ruta óptima.	60
2.2.1.	Diseño de grafos.	60
2.2.2.	Codificación.....	60
2.3.	Desarrollar el algoritmo que permita trazar una ruta óptima ante congestionamiento vehicular o accidentes.	62
2.4.	Realizar las pruebas de trazado de ruta, se utiliza el mapa en base a la ubicación de los semáforos.	65
3.	TERCERA FASE: Integrar y simular agentes inteligentes y algoritmo de rutas óptimas para evitar aglomeraciones y accidentes.	69
3.1.	Diseño de la interfaz del sistema.	69
3.2.	Realizar pruebas de funcionalidad.....	72
3.3.	Realizar pruebas de factibilidad del sistema.....	102
3.4.	Presupuesto Implementación.....	103
3.4.1.	Topología física y lógica propuesta.	104

3.4.2. Características de la alternativa con fibra óptica.....	105
3.4.3. Características de la alternativa 2 con puntos de acceso.	110
3.4.4. Ventajas y Desventajas de las alternativas propuestas	112
g. Discusión	114
1. Desarrollo De La Propuesta Alternativa.....	114
2. Valoración técnica económica ambiental.....	116
2.1. Recursos.....	116
h. Conclusiones	121
i. Recomendaciones	122
j. Bibliografía	123
k. Anexos.....	127
ANEXO 1. Informe de la Ubicación de los Semáforos.....	127
ANEXO 2. Artículo Científico.	135
ANEXO 3. Certificación Traducción Summary.....	157

Índice de Figuras

FIGURA 1. TARJETA RFID	17
FIGURA 2. MODELO DE COMUNICACIONES EN FIPA/ACL.	24
FIGURA 3. ARQUITECTURA DELIBERATIVA.	25
FIGURA 4. PLATAFORMA Y ARQUITECTURA JADE.	30
FIGURA 5. PLATAFORMAS JADE COMUNICÁNDOSE A TRAVÉS DE LA RED/NUBE.....	31
FIGURA 6. GRAFOS SIMPLES.	32
FIGURA 7. GRADO DE LOS VÉRTICES EN UN GRAFO NO DIRIGIDO Y DIRIGIDO.	35
FIGURA 8. GRADO DE LOS VÉRTICES EN UN GRAFO NO DIRIGIDO Y DIRIGIDO.	35
FIGURA 9. ESTRUCTURA DE LOS AGENTES DELIBERATIVOS.....	40
FIGURA 10. MODELO DE AGENTES.....	49
FIGURA 11. ESTABLECER COMUNICACIÓN ENTRE AGENTES.	50
FIGURA 12. INFORMAR NUEVO AGENTE.	50
FIGURA 13. INFORMAR CONGESTIONAMIENTO.....	51
FIGURA 14. CONTENEDOR PRINCIPAL DE JADE.	52
FIGURA 15. INICIO DE AGENTES.	52
FIGURA 16. CLASE EMISORRECEPTOR.....	53
FIGURA 17. CÓDIGO JAVA ENVIÓ MENSAJES SEMÁFORO MÁS CERCANO.	54
FIGURA 18. CÓDIGO JAVA RECEPCIÓN DE MENSAJES.	55
FIGURA 19. MENSAJE PARA INFORMAR UN ACCIDENTE.	56
FIGURA 20. MENSAJE PARA INFORMAR UN CONGESTIONAMIENTO.	57
FIGURA 21. MÉTODO AGREGAR MARCADORES.....	58
FIGURA 22. NUEVO AGENTE AGREGADO.....	58
FIGURA 23. MENSAJES CUANDO SE AGREGA UN NUEVO AGENTE.....	59
FIGURA 24. CÓDIGO AGENTE MÁS CERCANO AL USUARIO.	61
FIGURA 25. CÓDIGO AGENTE MÁS CERCANO AL DESTINO DEL USUARIO.	61
FIGURA 26. RECONFIGURAR RUTA.	62
FIGURA 27. FUNCIONAMIENTO GRAFO.....	63
FIGURA 28. DISEÑO DEL ALGORITMO.	64
FIGURA 29. UBICACIÓN EVENTOS.	65
FIGURA 30. RESULTADO PRUEBA.....	66

FIGURA 31. UBICACIÓN EVENTOS 2.....	67
FIGURA 32. RESULTADO PRUEBA 2.....	68
FIGURA 33. VENTANA PRINCIPAL DEL SISTEMA.....	69
FIGURA 34. CREAR AGENTE.....	70
FIGURA 35. PRODUCIR EVENTO (ACCIDENTE O CONGESTIONAMIENTO).....	71
FIGURA 36. SIMULAR RUTA.....	72
FIGURA 37. PRUEBA 1.....	73
FIGURA 38. PRUEBA 2.....	74
FIGURA 39. PRUEBA 3.....	75
FIGURA 40. PRUEBA 4.....	76
FIGURA 41. PRUEBA 5.....	77
FIGURA 42. PRUEBA 6.....	78
FIGURA 43. PRUEBA 7.....	79
FIGURA 44. PRUEBA 8.....	80
FIGURA 45. PRUEBA 9.....	81
FIGURA 46. PRUEBA 10.....	82
FIGURA 47. PRUEBA 11.....	83
FIGURA 48. PRUEBA 12.....	84
FIGURA 49. PRUEBA 13.....	85
FIGURA 50. PRUEBA 14.....	86
FIGURA 51. PRUEBA 15.....	87
FIGURA 52. PRUEBA 16.....	88
FIGURA 53. PRUEBA 17.....	89
FIGURA 54. PRUEBA 18.....	90
FIGURA 55. PRUEBA 19.....	91
FIGURA 56. PRUEBA 20.....	92
FIGURA 57. PRUEBA 21.....	93
FIGURA 58. PRUEBA 22.....	94
FIGURA 59. PRUEBA 23.....	95
FIGURA 60. PRUEBA 24.....	96
FIGURA 61. PRUEBA 25.....	97
FIGURA 62. PRUEBA 26.....	99

FIGURA 63. PRUEBA 27.....	100
FIGURA 64. PRUEBA 28.....	101
FIGURA 65. RESULTADOS.....	103
FIGURA 66. TOPOLOGÍA EN ANILLO.....	104
FIGURA 67. Tendido de Fibra óptica.....	105
FIGURA 68. Manga de Empalme.....	106
FIGURA 69. Gráfico de instalación de Manga de Empalme.....	106
FIGURA 70. ODF plástico.....	107
FIGURA 71. Pigtail de fibra óptica.....	107
FIGURA 72. Ejemplo de Internet inalámbrico en una ciudad.....	110
FIGURA 73. Vista de semáforos funcionando con Access Point.....	112

Índice de Tablas

TABLA I. ESTABLECER COMUNICACIÓN ENTRE AGENTES.....	44
TABLA II. INFORMAR CONGESTIONAMIENTO ENTRE AGENTES.....	45
TABLA III. INFORMAR ACCIDENTE.....	45
TABLA IV. ESTABLECER COMUNICACIÓN ENTRE AGENTES.....	46
TABLA V. INFORMAR CONGESTIONAMIENTO.....	47
TABLA VI. INFORMAR ACCIDENTE.....	47
TABLA VII. RESULTADOS PRUEBAS.....	102
TABLA VIII. PRESUPUESTO ALTERNATIVA 1.....	108
TABLA XI. RECURSOS MATERIALES.....	118
TABLA XII. RECURSOS DE HADWARE.....	118
TABLA XIII. RECURSOS DE SOFTWARE.....	119
TABLA XII. RECURSOS DE COMUNICACIÓN.....	119
TABLA XIII. COSTO TOTAL.....	120
TABLA XIV. LUZ INCANDESCENTE.....	127
TABLA XV. LUZ LED.....	132

c. Introducción

El incremento [1] del parque automotriz que se produce cada año en el país, hace que la parte céntrica de las ciudades del país se produzca congestionamientos y con ellos posibles accidentes, al mismo tiempo la contaminación que produce al medio ambiente.

El desarrollo de sistemas inteligentes para el control de tráfico automotriz, permite controlar el tráfico más eficaz y eficientemente, se controla en tiempo real detectando accidentes, verificando si existe aglomeración de vehículos o de personas y ayudando a los conductores a elegir rutas alternas para poder llegar a su destino evitando contratiempos, permitiendo el ahorro de combustible, control de tiempo por parte del conductor y mayor fluidez en el tráfico vehicular y peatonal.

En la actualidad los semáforos son sistemas temporizados para cambiar de un estado a otro y siguen un patrón de secuencia fija, ya que carece de inteligencia para tomar decisiones; esto presenta una gran desventaja en las horas pico. Para ayudar a solventar el tráfico de manera más eficaz se propone utilizar semáforos inteligentes ya que de acuerdo al grado de congestión que presente alguna vía de una intersección prolongue un poco más el paso de los automóviles para así descongestionar las principales arterias viales.

Actualmente los semáforos en la ciudad en Loja [2], la mayoría de ellos son temporizados y algunos mecánicos, por esa razón se sincronizan en un tiempo específico para cambiar de estado “Rojo, verde, amarillo”.

Datos otorgados por la unidad municipal de transporte terrestre de la ciudad Loja a diciembre del 2014 actualmente existen 117 semáforos que se distribuyen alrededor de la ciudad de Loja.

- 38 semáforos con tecnología LED.
- 79 semáforos Incandescentes.

Al existir semáforos mecánicos y semáforos temporizados mezclados, estos no pueden sincronizarse, ni tampoco pueden comunicarse entre sí, imposibilitando que estos semáforos se adapten a las condiciones de tráfico en tiempo real, dando en ocasiones prioridad a calles vacías sin tráfico que calles que se encuentran saturadas.

El objetivo primordial de este proyecto es crear una arquitectura multiagente representado por semáforos que permitan comunicarse entre sí, desarrollado a través de un Framework denominado JADE y con estándares de comunicación FIPA-ACL, simulando accidentes o aglomeraciones los semáforos podrán tomar decisiones por si mismos e informar a sus semáforos más cercanos sobre un evento que se ha producido en su intersección.

El sistema multiagente también está diseñado para que el usuario evite pasar por los accidentes o aglomeraciones que existan, el usuario elije el punto de origen y el destino y se trazara una ruta evitando dicho eventos, ayudando a descongestionar las vías y haciendo el tráfico más fluido.

d. Revisión Literaria

1. Capítulo I: Semáforos.

Un semáforo es un dispositivo mecánico o electrónico, que regula el tráfico de vehículos y peatones en las intersecciones de caminos. El tipo más frecuente tiene tres luces de colores: verde, para avanzar; rojo, para detenerse; y amarillo señal preventiva y paso intermedio del verde al rojo [3], estos semáforos están ubicados generalmente en intersecciones para poder regular el flujo de vehículos y peatones.

Tiene como funciones principales.

- Alternar el permiso de circulación entre una corriente vehicular o peatonal y otras.
- Regular la velocidad de los vehículos para mantener la circulación continua.
- Controla la circulación por canales.
- Reducir el número de accidentes y colisiones, ocasionados generalmente en intersecciones.
- Genera orden el tráfico [4]

1.1. Tráfico vehicular en la ciudad de Loja.

En los últimos años el tráfico vehicular en la ciudad de Loja aumento de una manera vertiginosa, según un estudio realizado por el municipio de Loja y el sindicato de choferes de la ciudad de Loja demuestra que tráfico vehicular se ha duplicado, tomando como referencia hace 10 años que el número de vehículos era de 11.500 y en la actualidad es de 24.000 aproximadamente [5].

El tráfico vehicular y la desincronización de semáforos agravan el problema de flujo vehicular, causando congestionamiento en determinados tramos y en horas pico haciendo imposible el flujo vehicular [6]. A pesar que en la ciudad de Loja se implementaron alrededor de 200 semáforos nuevos en 200 intersecciones estos semáforos se desprograman, se paralizan o se apagan causando molestia en conductores y peatones [7].

1.2. Semáforos Inteligentes.

En la mayoría de países en vías de desarrollo cuentan con semáforos temporizados que pasan de un estado a otro rigiéndose en un determinado tiempo, carecen de inteligencia para tomar decisiones, lo que representa una gran desventaja durante las horas picos en importantes arterias viales ya que los cambios se realizan en tiempos no adaptados a las condiciones del tráfico, y mientras una intersección vacía tiene luz verde la arteria principal se detiene a esperar el cambio, agrupando los vehículos hasta congestionar el canal [8].

Es considerado semáforo inteligente a todo aquel capaz de tomar decisiones dependiendo de una serie de parámetros de entrada (flujo de vehículos, velocidad media, identificar la calle, entre otros). Estos semáforos se presentan para solucionar varios problemas de tránsito comunes en todo el mundo [8, p. 2], por ejemplo:

- Congestión vehicular.
- Espera innecesarias.
- Mayor gasto de combustible.
- Mayor contaminación del medio ambiente.

La ventaja de los semáforos inteligentes es que permite optimizar la gestión de tráfico, de esta forma se ha comprobado que se reduce un 28% el tiempo de espera en los cruces en horas pico y un 6.5 % las emisiones de CO2 [9].

1.2.1. Tipos de semáforos inteligentes.

Existen diferentes formas de implementar semáforos con inteligencia, y esto puede abarcar desde una simple decisión del estado en que debe permanecer (luz, verde o roja) de acuerdo al tránsito hasta una decisión de mayor jerarquía, como la de tomar una decisión a causa de un accidente, cambiar el flujo etc.

1.2.1.1. Semáforos inteligentes con RFID.

La idea principal de este semáforo inteligente es de controlar el tráfico, haciendo variar los tiempos de espera en los puntos de intersección de acuerdo a la congestión vehicular. El

sistema consta de 4 partes principales, la tarjeta RFID [10], punto de acceso, ubicación de servidor de redes, WAN y una base centraliza que almacena los datos y a partir de ellos escoge una alternativa.

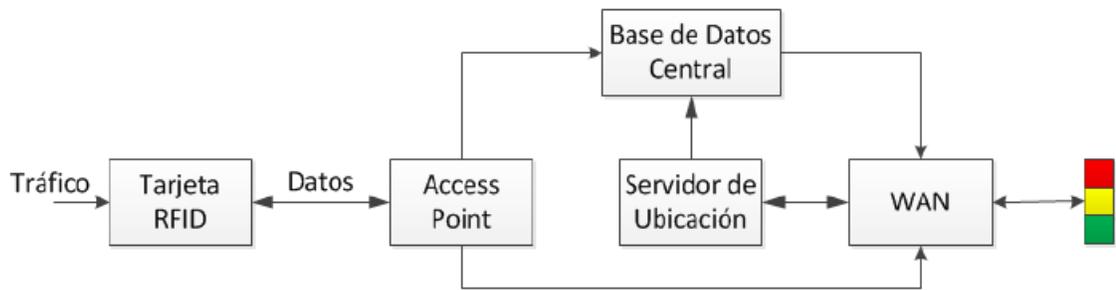


FIGURA 1. Tarjeta RFID

1.2.1.2. Semáforo Inteligente usando Redes de sensores Inalámbricos.

Capaz de controlar el tráfico y manipular en forma dinámica los tiempos de espera. Este sistema cuenta con 2 partes principales. La red de sensores inalámbricos (Wireless Sensor Network o WSN) y la estación base (Base Station o BS), que se encarga de ejecutar los algoritmos de control.

La red de sensores inalámbricos, consiste en un grupo de sensores diseñados para proporcionar la infraestructura de comunicación de tráfico y facilitar el flujo de tránsito. Cada sensor se encarga de generar los datos de tráfico como el número de vehículos, los procesos de salida, velocidad de cada vehículo, su longitud. Los datos recopilados se envían en tiempo real a la base.

1.2.1.3. Semáforos Inteligentes mediante Procesamiento de Imágenes.

Mediante el control de tráfico utilizando procesamiento de imágenes, es posible medir la intensidad del tráfico y de acuerdo a los datos obtenidos modificar los tiempos del semáforo.

El funcionamiento es de la siguiente manera:

- Adquisición de Imágenes. Se captura la carretera vacía y la imagen con el tráfico actual. La imagen de carretera vacía se guarda como una imagen de referencia.
- Se convierte ambas imágenes RGB a la escala grises.
- Mejora la imagen.
- Verificar coincidencia entre la imagen de referencia y la imagen con el tráfico actual. Se utiliza el método de detección de borde morfológica porque requiere menos cálculo computacional y también es capaz de extraer los bordes independientemente de su dirección.

Después de un procedimiento de detección de bordes de ambas imágenes, los semáforos se pueden controlar basándose en el porcentaje de coincidencia.

1.2.1.4. Semáforo Inteligente basado en Inteligencia Artificial.

Para el sistema de semáforo inteligente, permite la aplicación de las reglas de la vida real similar a la manera en que nosotros pensamos.

Por ejemplo si una calle esta congestionada debido a un accidente, el semáforo inteligente debe de avisar al conductor que dicha calle esta congestionada y que debe de tomar otra vía alterna para evitar el congestionamiento.

El objeto de los semáforos Inteligentes con Inteligencia Artificial es que los semáforos se logren comunicar entre sí, de tal forman que interactúen de acuerdo a su necesidad. Los principales objetivos para la creación de Semáforo Inteligente con I.A son:

- Avisar a conductores. En caso de que exista congestionamiento, estos deberán tomar rutas alternas.
- Comunicación entre semáforos. Que los semáforos interactúen entre sí, de acuerdo a la necesidad que estos tienen y que esta comunicación sea en tiempo real.
- Capaz de toma decisiones de una manera independiente.

- Forzar semáforos para condiciones especiales como: (paso de un policía, ambulancias, bomberos).
- Procesar datos. Analizar las prioridades de las avenidas y dar paso a la de mayor preferencia.

2. Capítulo II: Sistemas Multiagentes.

La necesidad que tiene el ser humano de interactuar con tecnología que lo rodea, como computadoras, dispositivos móviles, etc. Hace que la Inteligencia Artificial actúe de forma directa implementando técnicas y metodologías para que dichas tecnologías interactúen de forma dinámica con una persona.

2.1. Definición de Agentes.

Existen varias definiciones de Agentes, entre ellas está la de FIPA [11] (Foundation for Intelligent Physical Agents): “Un agente es una entidad de software encapsulado con su propio estado, conducta, hilo de control y la habilidad para interactuar y comunicarse con otros entidades (gente, otros agentes o sistemas)”.

Entre otra definición esta: Un agente es un sistema computacional capaz de actuar de manera autónoma para satisfacer sus objetivos y metas, mientras se encuentra situado persistentemente en su medio ambiente [12].

Según Wooldridge & Jennings, “Un agente Inteligente debe ser capaz de ejecutar flexiblemente acciones autónomas con el fin de cumplir sus objetivos”, donde flexibilidad significa.

- Reactividad. Los agentes Inteligentes deben ser capaces de percibir su medio ambiente y responder a tiempo a los cambios en el, a través de sus acciones.
- Pro actividad. Capacidad de exhibir un comportamiento dirigido a objetivos, tomando la iniciativa.
- Habilidad social. Capacidad de interactuar con otros agentes (y posiblemente humanos) a través de un lenguaje de comunicación.

2.2. Sistemas Multiagente.

Sistemas Multiagente son sistemas compuestos por una serie de entidades de software autónomos llamados agentes distribuidos. En teoría, los sistemas Multiagente se caracterizan generalmente en términos de comportamientos internos y las interacciones entre los agentes externos.

Representa un poderoso modelo para resolver problemas de computación distribuidos, además de ser capaces de adaptar su funcionamiento en entornos abiertos y dinámicos en los que el contenido y la carga de trabajo están cambiando continuamente.

Los sistemas multiagente son capaces de cooperar con otros agentes para resolver problemas, esto se da cuando agentes individuales no quieren o no pueden realizar tareas dentro de ciertas limitaciones o no tienen la capacidad para realizar tareas por sí mismos, los agentes pueden cooperar usando un lenguaje de comunicación de agentes (ACL) [13].

Capacidad de los sistemas Multiagente [14].

- Descomposición del problema.
- Comunicación entre agentes.
- Coherencia de actuaciones.
- Representación del conocimiento a otra agentes.
- Coordinación y acciones.
- Gestionar el uso de recursos.
- Evitar comportamientos no deseados.
- Diseño del MAS (Sistemas Multiagente): metodología y desarrollo.

2.3. Comunicación entre agentes.

La comunicación entre agentes es la llave para obtener todo el potencial del paradigma de agentes, al igual que el desarrollo del lenguaje humano fue la llave para el desarrollo de la inteligencia y de la sociedad.

Los mensajes deben poder ser.

- Servicios por eventos.
- Síncronos o asíncronos.
- Direcciones físicas o por rol
- Unicast / multicast / broadcast.

Los lenguajes de comunicación deben tener una semántica bien definida, ya que los sistemas multiagente están orientados a la resolución distribuida de problemas y los agentes solo pueden resolver estos problemas si existe una cooperación entre agentes.

2.4. Modelos de comunicación.

En los sistemas multiagente, el lenguaje de comunicación es indispensable, ya que mediante la comunicación puede existir la cooperación entre agentes y poder resolver problemas.

Existen los modelos de comunicación, entre los que está.

Arquitectura de Pizarra. [15].

- Zona de trabajo común en la que se encuentra la información a compartir.
- No hay comunicación directa entre los agentes.
- Pueden existir agentes con tarea de control específicas.
- Pueden existir varias pizarras.
- Inconveniente: comunicación centralizada.

Paso de mensajes.

- La comunicación se establece directamente entre dos agentes (emisor y receptor).
- Pueden existir agentes facilitadores.
- Ventaja: Flexibilidad.

Sin comunicación. Los agentes pueden interactuar sin conocerse, infiriendo las otras intenciones de otros agentes, esto puede deberse a la imposibilidad técnica o al deseo de una mayor autonomía por parte de los agentes.

2.5. Protocolos de información.

Representan los patrones que modelan las posibles comunicaciones, los que participan en la conversación deben conocer el protocolo comunicación. Entre las implementaciones del protocolo de información esta FIPA-ACL.

2.5.1. FIPA-ACL.

FIPA es una organización internacional dedicada a la promoción de la industria de los agentes inteligentes mediante el desarrollo de especificaciones que soporten la interoperabilidad entre agentes y aplicaciones basadas en agentes.

Cubre todos los aspectos de un entorno de agentes.

- Aplicaciones.
- Arquitectura.
- Comunicación.
- Gestión de agentes.
- Transporte de mensajes.

El agente es capaz de enviar un mensaje, codificarlo para su transmisión como una secuencia de bytes. En el caso más habitual, el servicio es.

- De confianza (Los lenguajes bien formados llegan a su destino).
- Fiable (el mensaje se recibe tal y como se envía).
- Ordenado.

El servicio de envío de mensajes detecta condiciones de error, como mensaje mal formado, no-enviable, agente no alcanzable, etc. Un agente tendrá un nombre tal que permita al servicio de envío de mensajes remitir el mensaje a la dirección correcta. El servicio de envío de mensaje debe ser capaz de determinar el mecanismo de transporte correcto (TCP/IP, SMTP, HTTP, etc...) y permitir cambios en la ubicación del agente si es necesario.

2.6. Teoría de los actos del habla.

Es la base teórica en la comunicación entre agentes, en un acto del habla se encuentran tres acciones.

- Locución. Contexto del hecho físico del acto del habla.
- Ilocución. Intención con la que se realiza.
- Perlocución. Acto que ocurre como resultado.

Objetivos de los actos del habla [14].

- Completitud. Cubrir un amplio rango de situaciones de comunicación.
- Simplicidad. No sobredimensionar el diseño de agentes simples.
- Concisión. Minimizar la redundancia y la ambigüedad para facilitar a los agentes la elección del acto comunicativo que deben emplear.

Requerimientos para la comunicación referente a FIPA-ACL [14].

- Los agentes deben enviar un mensaje “not-understood” si reciben un mensaje que no reconocen o cuyo contenido no son capaces de procesar, un agente debe estar preparado para recibir y manejar adecuadamente un mensaje “not-understood” de otros agentes.
- Un agente ACL puede escoger implementar cualquier subconjunto de tipos de mensajes y protocolos predefinidos, la implementación de estos mensajes debe ser correcta respecto a la definición semántica de los actos referenciados.
- Un agente ACL que utilice actos comunicativos cuyos nombres están definidos en esta especificación, deben implementarlos correctamente de acuerdo a su definición.
- Un agente ACL debe ser capaz de generar mensajes sintácticamente bien formados en la forma de transporte que corresponde al mensaje que desea enviar, de la misma forma, debe ser capaz de traducir una secuencia de caracteres que este bien formada en la sintaxis del mensaje correspondiente.

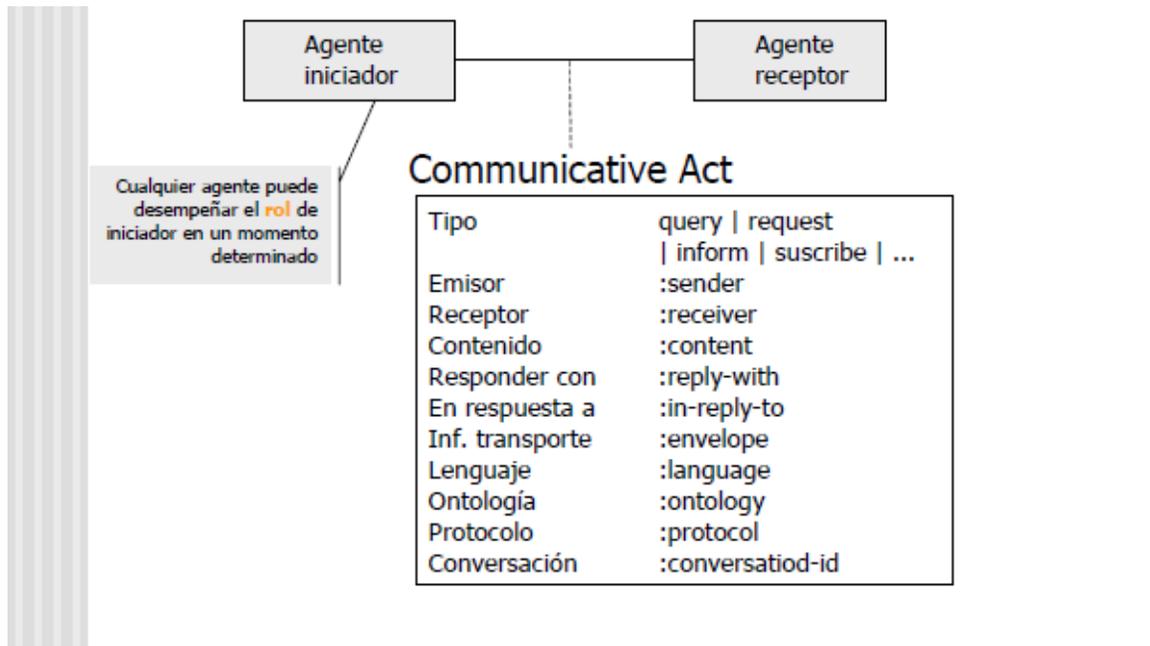


FIGURA 2. Modelo de Comunicaciones en FIPA/ACL.

2.7. Arquitectura de Agentes Inteligentes.

Existen diferentes tipos de arquitecturas, reactivas, deliberativas, híbridas. A continuación se detalla las siguientes arquitecturas.

2.7.1. Arquitectura deliberativa.

La arquitectura deliberativa también conocida como BDI (Belief-Desire-Intention)

En la arquitectura DBI el agente es visto como un agente racional con un conjunto de actitudes mentales.

- Creencias.
- Deseos
- Intenciones.

El agente toma sus acciones en función de sus estados, esta aproximación es considerada como muy apropiada para agentes en entornos reales, complejos y dinámicos.

BDI (Creencias, Deseos, Intenciones) [16].

- Creencia. Una creencia del agente es la información que tiene el agente sobre el entorno.
- Deseo. Un deseo del agente es algo que el agente quiere conseguir y que por lo tanto tiene preferencia. Los objetivos es un subconjunto de deseos consistentes entre si que el agente debe perseguir.
- Intenciones. Las intenciones del agente son los objetivos establecidos. Existe un compromiso por parte del agente a conseguir sus intenciones.

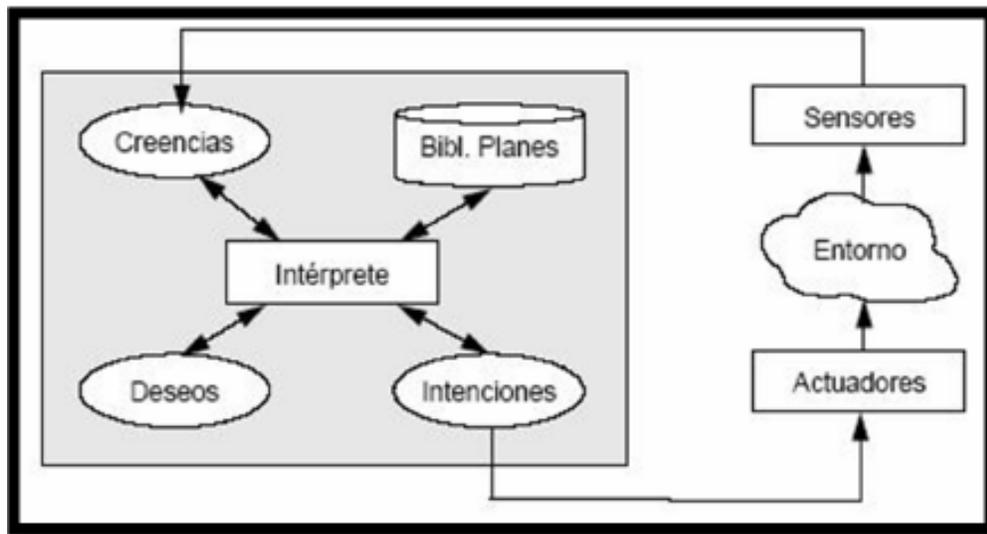


FIGURA 3. Arquitectura Deliberativa.

En la figura se observa la arquitectura deliberativa, donde los sensores obtienen la respuesta del entorno en el cual hace cambiar la creencia del agente, esta creencia ejecuta un objetivo o deseos, los deseos u objetivos, ejecutan ciertos planes para poder cumplir intenciones que son objetivos a corto plazo. Una vez cumplidos los objetivos a corto plazo son enviados a los actuadores.

2.7.2. Arquitecturas reactivas.

Los agentes reactivos se caracterizan porque pueden operar rápida y efectivamente sin la necesidad de procesar una representación simbólica del entorno, ya que representan una categoría de agentes que no posee modelos simbólicos externos del entorno en el que se encuentran. Los agentes reactivos toman decisiones basadas totalmente en el presente, sin hacer uso de lo que ha ocurrido en el pasado, porque no conocen su historia. La característica más importante de los agentes reactivos es el hecho de que los agentes son relativamente simples e interaccionan con otros agentes de manera sencilla

2.7.3. Arquitectura Híbrida.

La arquitectura consiste en los subsistemas de percepción y acción, contando con diferentes capas. Estas capas pueden ser reactivas y deliberativas, las cuales son independientes de su acción.

2.8. Metodologías para el desarrollo de agentes inteligentes.

Un proceso de software detallado y completo suele denominarse metodología. Las metodologías se basan en una combinación de los modelos de proceso genéricos. Adicionalmente una metodología debería definir con precisión los artefactos, roles y actividades involucrados, junto con prácticas y técnicas recomendadas, guías de adaptación de la metodología al proyecto, guías para uso de herramientas de apoyo, etc. [17].

Elementos que forman parte de una metodología [18].

- Fases. Tareas a realizar en cada fase.
- Productos. Entrada/ Salida de cada fase, documentos.
- Procedimientos y herramientas. Apoyo a la realización de cada tarea.
- Criterios de evaluación. Del proceso y del producto. Saber si se ha logrado los objetivos.

Actualmente existen un gran número de metodologías propuestas, lo que dificulta su evaluación y elección. Además, ya que las metodologías suelen ir acompañadas de herramientas de desarrollo y notaciones específicas, el concepto y las propiedades de los agentes diseñados, varía según la metodología que existe [19].

2.9. GAIA.

GAIA es una metodología para realizar los procesos de análisis y diseño orientado a agentes. Esta metodología hace uso de una serie de modelos de análisis y diseño.

2.9.1. Etapa de análisis.

En esta etapa el GAIA busca el entendimiento del sistema y su estructura como una organización. Una organización puede ser vista como un conjunto de roles que mantienen ciertos tipos de interacción con otros roles del sistema. Un rol se define en GAIA por 4 atributos.

1. Las responsabilidades. Las responsabilidades pueden fraccionarse en dos categorías: las propiedades de vida y las propiedades de seguridad, las responsabilidades determinan la funcionalidad y son tal vez los atributos claves asociados a un rol.
2. Los permisos. Los permisos identifican los recursos que el rol tiene disponibles para realizar sus responsabilidades
3. Las actividades. Las actividades
4. representan acciones privadas u operaciones que el agente puede completar sin la necesidad de interactuar con otros agentes en el sistema.
5. Protocolos. Los protocolos definen las formas en que un agente asumiendo dicho rol podría interactuar con otros agentes.

Durante el análisis con GAIA se generan dos tipos de modelos: de roles y de interacción [20].

2.9.1.1. El Modelo de Roles.

Permiten identificar los distintos tipos de roles que se encuentran en el sistema. Con cada rol hay asociados unos permisos y unas responsabilidades. Las responsabilidades son de dos tipos, de vida y de seguridad. Con estos conceptos, se puede especificar un esquema de rol para cada tipo de rol en la organización y conformar así el modelo de roles. Un esquema de rol presenta una descripción del rol, sus protocolos y actividades, sus permisos y las responsabilidades de vida y seguridad.

2.9.1.2. El Modelo de Interacción.

Es una serie de definiciones de protocolo, una para cada tipo distinto de interacción entre los roles. Cada definición de protocolo consiste de un propósito de la interacción, los roles iniciador y receptor de la misma, las entradas y salidas y el procesamiento ejecutado.

En la fase de diseño se construyen los modelos de agentes, modelo de servicios y modelo de conocimiento (modelo de relación).

2.9.1.3. El modelo de agentes.

Define los tipos de agentes y las instancias que se crearán en la ejecución. Los tipos de agentes se crean por agregación de roles. La notación empleada, en forma de árbol, permite definir los tipos de manera jerárquica: si un tipo está compuesto por varios subtipos, es que está compuesto por los roles incluidos en cada subtipo.

2.9.1.4. El modelo de servicios.

Identifica los servicios asociados a cada rol, y las propiedades de estos servicios.

En GAIA, un servicio es una función de un agente. Cada servicio se define mediante las siguientes propiedades:

- Entradas
- Salidas
- Pre-condiciones
- Post-condiciones

2.9.1.5. El modelo de conocimiento o de relaciones.

Muestra los flujos de comunicación entre tipos de agentes, sin detallar los mensajes intercambiados, utilizando la notación común de grafos, en los que los nodos representan los tipos de agente.

La metodología GAIA se considera una metodología fácil de aprender, donde la mayor complejidad radica en la notación formal empleada, es una metodología apta para sistemas pequeños y medianos, ya que no aporta soluciones para tratar con un gran número de agentes, y que se mantiene a un nivel de abstracción que permite la implementación en cualquier plataforma. Por la misma razón, no ofrece guía ni herramientas que faciliten la fase de implantación. Tampoco proporciona mecanismos de verificación de consistencia sobre los modelos desarrollados.

2.10. JADE

Jade (Java Agente Development Framework) [21] es un Framework desarrollado totalmente en Java, agiliza el desarrollo de agentes inteligentes, sistemas multiagentes de una manera más eficaz y eficientes, esto a través de un middleware que cumple con las especificaciones y estándares FIPA (Foundation for Intelligent Physical Agents) y con un lenguaje estandarizado para la comunicación entre agentes llamado FIPA-ACL (Foundation for Intelligent Physical Agents – Agent Communication Language).

FIPA-ACL [22] Es el protocolo que utilizan los agentes para el intercambio de mensajes, el servicio de transporte es capaz de enviar un mensaje, codificarlo para su transmisión como una secuencia de bytes.

El servicio de mensaje es de confianza, ya que los mensajes bien estructurados llegan a su destino, es fiable ya que el mensaje se recibe tal y como se envía, recibiendo el mensaje ordenadamente.

El servicio de mensaje es capaz de detectar condiciones de error al momento de enviar un mensaje, como mensaje mal formado, mensaje imposible de enviar, agente no alcanzable.

FIPA-ACL especifica que cada agente debe tener un nombre como identificación, tal que permita al servicio de envío de mensajes remitir el mensaje a la dirección correcta, el servicio de mensajes debe ser capaz de determinar el mecanismo de transporte correcto (TCP/IP, HTTP, etc...) y permitir cambios en la ubicación del agente si es necesario.

2.10.1. Plataforma y Arquitectura de JADE.

JADE es una plataforma de agentes distribuida. Los agentes se implementan con hilos en java y se almacenan dentro de un Agent Containers (Contenedor de Agentes) que proporciona el soporte para la ejecución del agente. Los mensajes se transfieren codificados como objetos Java, en lugar de cadenas, con el fin de evitar la clasificación (marshalling) y la desclasificación (unmarshalling) de procedimientos. Al cruzar los límites de la plataforma, el mensaje se convierte automáticamente a / desde la sintaxis compatible con FIPA, codificación y protocolo de transporte.

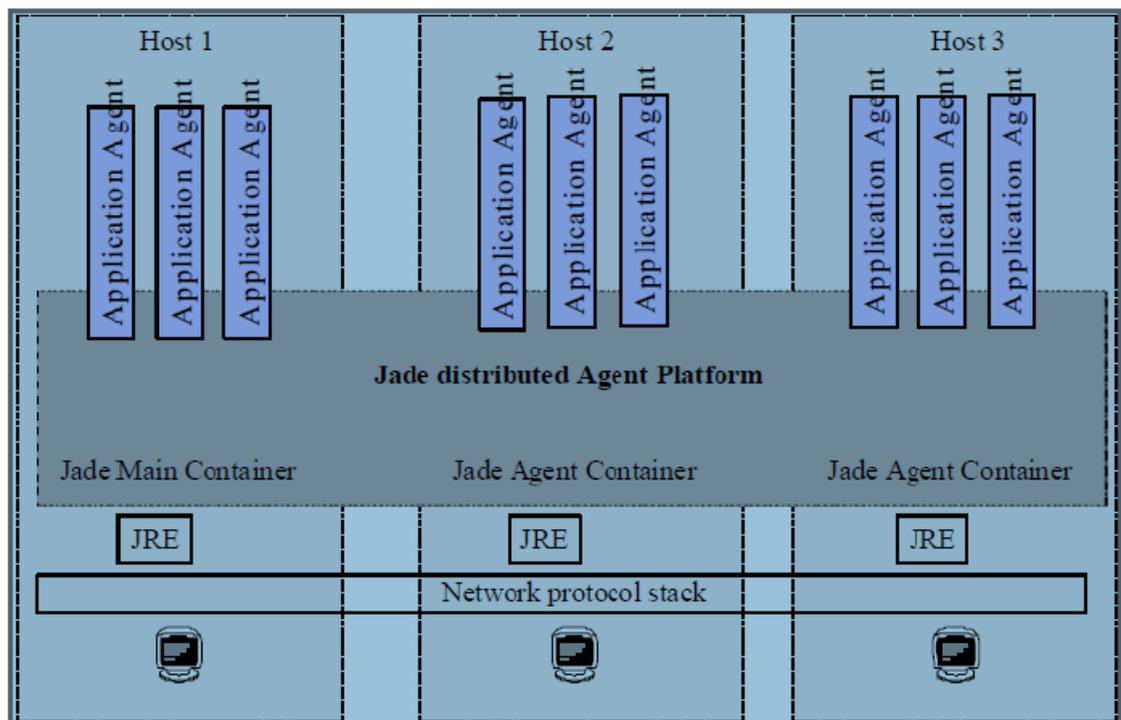


FIGURA 4. Plataforma y arquitectura JADE.

Jade cumple plenamente con esta arquitectura de referencia y cuando se pone en marcha la plataforma JADE se crea inmediatamente un contenedor principal (Main Container). Además el servicio de mensajería está siempre activada para permitir la comunicación basada en mensajes.

Cada instancia de la plataforma JADE se llama contenedor la cual puede albergar a distintos agentes, así mismo un conjunto de contenedores que se encuentren activos forman una plataforma.

En la siguiente figura se representa dos plataformas JADE comunicándose a través de la red/nube.

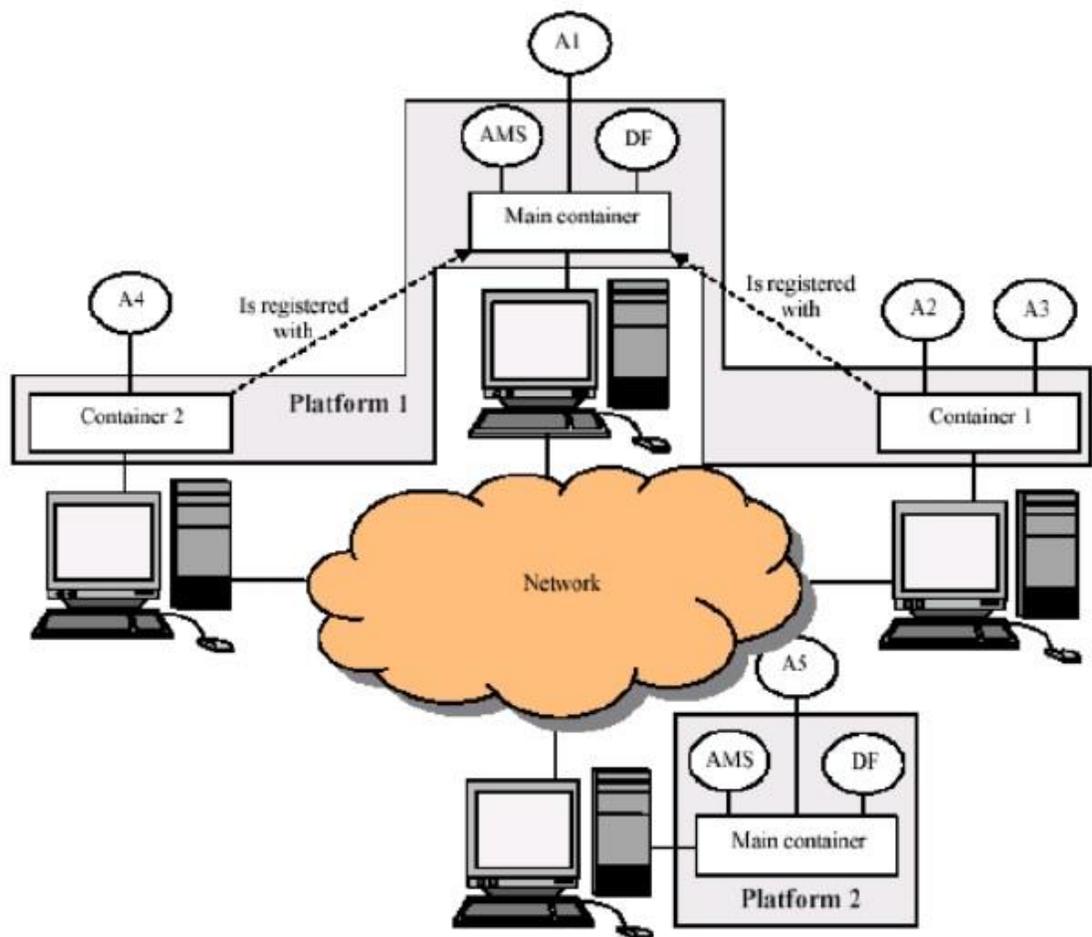


FIGURA 5. Plataformas JADE comunicándose a través de la RED/NUBE.

3. Capítulo III: GRAFOS

3.1. Introducción.

El origen de la palabra grafo es griego y su significado etimológico es "trazar". Aparece con gran frecuencia como respuesta a problemas de la vida cotidiana, algunos ejemplos podrían ser los siguientes: un gráfico de una serie de tareas a realizar indicando su secuenciación (un organigrama), grafos matemáticos que representan las relaciones binarias, una red de carreteras, la red de enlaces ferroviarios o aéreos o la red eléctrica de una ciudad. En cada caso, es conveniente representar gráficamente el problema dibujando un grafo como un conjunto de puntos fig1. Grafos simples (vértices) con líneas conectándolos (arcos) [23].

El desarrollo de los grafos de ha estructurado en base a dos grandes objetivos:

- Desarrollo de un interfaz para la construcción y edición de grafos en modo tabular o gráfico, que permita la incorporación modular de multitud de funciones.
- Desarrollo de una estructura de clases y librerías .dll con algoritmos de resolución y análisis de problemas de teoría de grafos. Implementación y compilación de diferentes algoritmos.

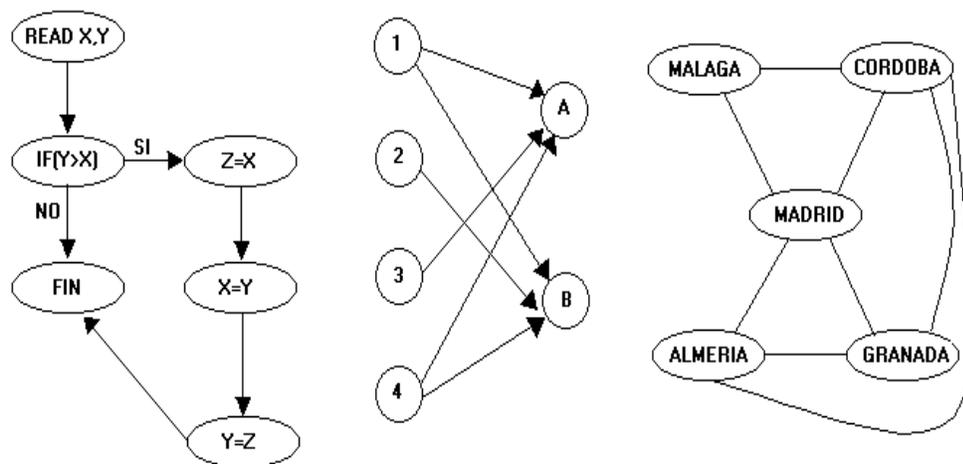


FIGURA 6.Grafos simples.

3.2. Terminología de grafos [24]:

- Vértice: Nodo.
- Enlace: Conexión entre dos vértices (nodos).
- Adyacencia: Se dice que dos vértices son adyacentes si entre ellos hay un enlace directo.
- Vecindad: Conjunto de vértices adyacentes a otro.
- Camino: Conjunto de vértices que hay que recorrer para llegar desde un nodo origen hasta un nodo destino.
- Grafo conectado: Aquél que tiene camino directo entre todos los nodos.
- Grafo dirigido: Aquél cuyos enlaces son unidireccionales e indican hacia donde están dirigidos.
- Grafo con pesos: Aquél cuyos enlaces tienen asociado un valor. En general en este tipo de grafos no suele tener sentido que un nodo se apunte a sí mismo porque el coste de este enlace sería nulo.

3.3. DEFINICIONES

- **Definición 1.** Un grafo se define como un par $G = (V, A)$, donde V es un conjunto finito no vacío de vértices y A es un conjunto de pares de vértices de V , es decir, las aristas.
- **Definición 2.** Un grafo G se define como un par ordenado, $G = (V, A)$, donde V es un conjunto finito y A es un conjunto que consta de dos elementos de V .

3.4. TERMINOLOGÍA y CONCEPTOS

3.4.1. Grafos dirigidos y no dirigidos

Dependiendo del tipo de relación entre los vértices del grafo, se definen distintos tipos de grafos. Así se distinguen aristas dirigidas y no dirigidas:

- **Arista dirigida:** es aquella que define un par ordenado de vértices (u,v) , donde el primer vértice u es el origen de la arista y el segundo vértice v es el término (o vértice final). El par $(u, v) \neq (v, u)$.

- **Arista no dirigida:** es aquella que define un par no ordenado de vértices (u, v) , donde $(u, v) = (v, u)$.

De esta forma se distinguen entre grafos dirigidos y grafos no dirigidos.

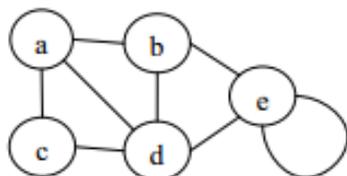
- **Grafo dirigido:** Es aquel cuyas aristas son dirigidas. Los grafos dirigidos suelen representar relaciones asimétricas como por ejemplo: relaciones de herencia, los vuelos entre ciudades, etc.
- **Grafo no dirigido:** Es aquel cuyas aristas son no dirigidas. Representan relaciones simétricas como relaciones de hermandad y colaboración, conexiones de transportes, etc.

3.4.2. Incidencia, adyacencia y grado de un vértice

Sea un grafo $G = (V, A)$, los vértices u y v pertenecientes a V ; y una arista (u,v) perteneciente a A , se dice que:

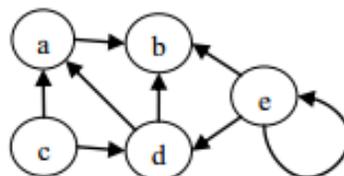
- **Incidencia:** la arista (u,v) es incidente con los vértices u y con v .
- **Adyacencia:** Dos vértices u y v son adyacentes si existe una arista cuyos vértices sean u y v :
 - El vértice u es adyacente a v
 - El vértice v es adyacente desde u
- **Grado:** El grado de un vértice u es el número de vértices adyacentes a u . Para un grafo dirigido, el grado de salida de un vértice u es el número de vértices adyacentes desde u , mientras que el grado de entrada de un vértice u es el número de vértices adyacentes a u .

Grafo no dirigido:



Grado (a) = 3
Grado (b) = 3
Grado (c) = 2
Grado (d) = 4
Grado (e) = 4

Grafo dirigido:



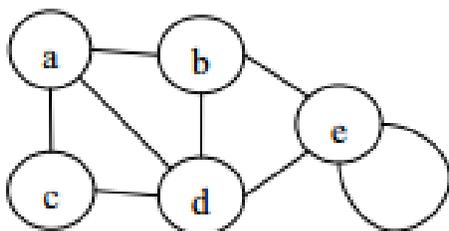
GradoE (a) = 2	GradoS (a) = 1
GradoE (b) = 3	GradoS (b) = 0
GradoE (c) = 0	GradoS (c) = 2
GradoE (d) = 2	GradoS (d) = 2
GradoE (e) = 1	GradoS (e) = 3

FIGURA 7. Grado de los vértices en un grafo no dirigido y dirigido.

3.4.3. Grafos simples y multígrafos

Un grafo simple es aquel que no tiene aristas paralelas o múltiples que unan el mismo par de vértices. Un grafo que cuente con múltiples aristas entre dos vértices se denomina multígrafo. La Figura 3 muestra un ejemplo de grafo simple y de multígrafo, donde existen aristas paralelas incidentes a los vértices a y d, y e y d. En este caso, se dice que el grafo tiene multiplicidad 2 (máximo de aristas paralelas entre dos vértices).

GRAFO SIMPLE:



MULTIGRAFO:

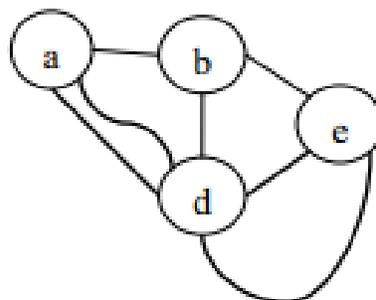


FIGURA 8. Grado de los vértices en un grafo no dirigido y dirigido.

3.4.4. Camino, bucle y ciclo

Un camino es una secuencia que alterna vértices y aristas que comienza por un vértice y termina en vértice tal que cada arista es incidente a su vértice predecesor y sucesor. Es decir, un camino es un sucesión de vértices de $V: \langle v_0, v_1, v_2, \dots, v_k \rangle$ que cumple que:

$$(v_i, v_{i+1}) \in A \quad \forall i \in \{0 \dots k-1\}.$$

Se dice que este camino tiene longitud k . Es decir, el número de aristas de un camino o ciclo es la longitud del camino.

Un camino es simple si cada vértice en el camino es distinto, excepto posiblemente por el primero y el último vértice. Un camino simple cumple la siguiente restricción:

$$v_i \neq v_j \quad \forall i \in \{0 \dots k\}, j \in \{1 \dots k-1\}, i \neq j$$

Para todo vértice x , existe el camino simple $\langle x \rangle$, que sería el camino de longitud 0.

Un bucle es un camino de longitud 1 que comienza y termina en el mismo vértice:

$$\langle x_i, x_i \rangle.$$

Un ciclo es un camino simple $\langle v_0, \dots, v_k \rangle$ que cumple las siguientes restricciones:

$$v_0 = v_k$$

Si es no dirigido, $k = 1$ (es un bucle) o $k \geq 3$.

3.5. Recorrido de un grafo:

Lo que queremos conseguir es pasar por todos los nodos de un grafo evitando tratarlos dos veces (por ejemplo, si se están escribiendo en pantalla el contenido de todos los nodos del grafo hay que evitar que un nodo se escriba dos veces). Para ello es importante usar o bien

un array de boolean que represente todos los nodos del grafo, o bien añadir un campo "Visitado" de tipo boolean dentro de cada nodo. De esta forma, cuando pasemos la primera vez por un nodo, pondremos su campo "Visitado" a True. Esto es necesario para saber si ya hemos pasado por un nodo y así poder evitar pasar de nuevo por él, pues de no saber si lo hemos visitado o no, el algoritmo podría entrar en un bucle infinito saltando siempre entre los dos mismos nodos.

Hay dos tipos de recorridos de un grafo, y pueden ser implementados tanto de forma recursiva como de forma iterativa. Estos recorridos son:

- En profundidad, que consiste en alejarse todo lo posible del nodo origen para después empezar a visitar los nodos restantes a la vuelta.

A lo ancho (o por nivel), que consiste en visitar primero los nodos vecinos del origen, luego los vecinos de éstos, y así sucesivamente.

e. Materiales y Métodos

1. Materiales

El desarrollo del sistema Multiagentes para la comunicación de semáforos, se realizó a través del lenguaje de programación Java, y utilizando el Framework JADE que proporciona el soporte adecuado para la creación de Agentes, se utilizó el gestor de BD MYSQL donde se almacenaran los semáforos y el mapa de la ciudad de Loja, a través de google maps donde se simulara accidentes y aglomeraciones. Se tomó en cuenta los semáforos que existen en la ciudad de Loja, con sus características y ubicación con coordenadas obtenidas por la unidad municipal de tránsito, del municipio de Loja.

Las fuentes de información a consultar se basaron en tesis, libros especializados en agentes inteligentes, sitios especializados de la temática a buscar (ver sección bibliográfica).

2. Métodos

Para el desarrollo del presente proyecto se utilizará una metodología que establezca una secuencia de pasos ordenados y lógicos, los cuales servirán de guía para la ejecución de diferentes actividades, alcanzando de esta manera los resultados que se plantearon al inicio de esta investigación.

a. Método científico.

Este método nos ayudó en la ejecución de todo el proyecto, debido a que nos proporciona el camino a seguir desde la observación del problema hasta la obtención de conclusiones, mediante la obtención y análisis de los factores que influyen en el proceso de capacitación.

b. Método Analítico.

Nos permitirá analizar de manera directa los semáforos, su funcionamiento, estructura y como dichos dispositivos ayuda a que el tráfico de una ciudad sea más ordenada.

c. Metodología de desarrollo de agentes Inteligentes (GAIA)

GAIA es una metodología para el diseño del sistema basada en agentes cuyo objetivo es obtener un sistema que maximice alguna medida de calidad global. GAIA pretende ayudar al analista a ir sistemáticamente desde unos requisitos iniciales a un diseño que, según los autores, este lo suficientemente detallado como para ser implementado directamente. GAIA propone trabajar inicialmente con un análisis de alto nivel. En este análisis se usan dos modelos, el modelo de roles para identificar los roles claves en el sistema junto con sus propiedades definitorias y el modelo de interacciones que define las interacciones mediante un referencia a un modelo institucionalizado de intercambio de mensajes, como el FIPA-ACL.

El objetivo de este diseño es generar tres modelos. El modelo de agentes que define los agentes que existen, cuantas instancias de cada tipo y que papeles juega cada agente, el modelo de servicios que identifica los servicios (funciones del agente) asociados a cada rol, y un modelo de conocidos, que define los enlaces de comunicaciones que existen entre los agentes.

f. Resultados

Para el desarrollo de trabajo de titulación se definió en tres fases, que ayudó al desarrollo eficiente y culminación exitosa del siguiente trabajo de titulación.

A continuación se detalla las actividades y resultados obtenidos por cada fase.

1. PRIMERA FASE: Desarrollar los agentes inteligentes para la comunicación de semáforos en tiempo real.

1.1. Estudio de los agentes deliberativos.

Los agentes deliberativos son aquellos que contienen un modelo simbólico del mundo, explícitamente representado, en donde las decisiones se toman utilizando mecanismos de razonamiento lógico basado en la concordancia de patrones y la manipulación simbólica.

Los agentes deliberativos poseen creencias, deseos e intenciones, estas características se conocen como BDI.

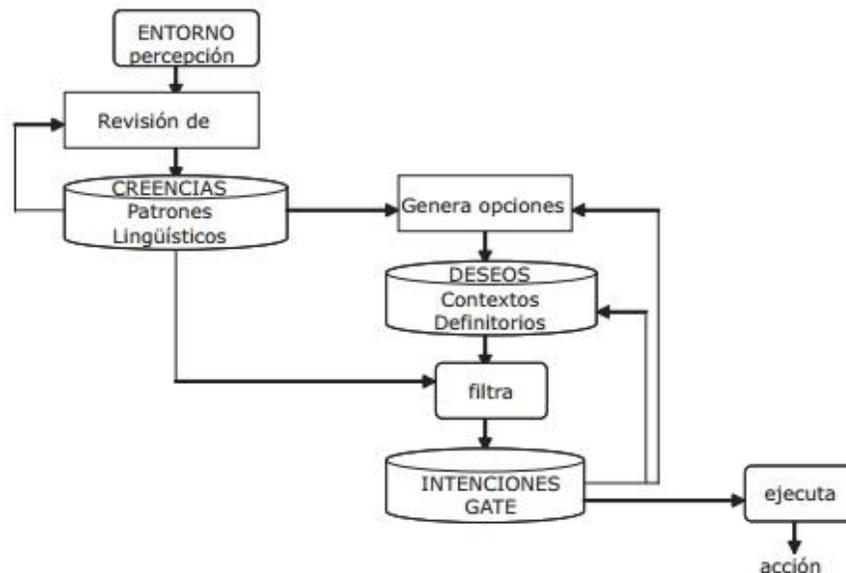


FIGURA 9. Estructura de los agentes deliberativos.

1.2. Selección de herramientas a utilizar para el desarrollo del sistema de agentes inteligentes

Para el desarrollo de sistema de agentes se utilizó varias herramientas tanto para crear, gestionar y depurar los agentes, así como para el análisis y diseño de los mismos guiándose en metodología creadas específicamente para el desarrollo de agentes y un sistema de gestión de base de datos para almacenar los agentes.

Se utiliza el lenguaje de programación Java (Lenguaje orientado a objetos) que se usa para el desarrollo de aplicaciones tanto de escritorio como aplicaciones web, así mismo el framework que facilita la creación de agentes llamado JADE esta hecho el cien por ciento en java, lo cual este lenguaje de programación será imprescindible para la realización del proyecto de titulación.

Jade (Java Agente Development Framework) es un Framework desarrollado totalmente en Java, agiliza el desarrollo de agentes inteligentes, sistemas multiagentes de una manera más eficaz y eficientes, esto a través de un middleware que cumple con las especificaciones y estándares FIPA (Foundation for Intelligent Physical Agents) y con un lenguaje estandarizado para la comunicación entre agentes llamado FIPA-ACL (Foundation for Intelligent Physical Agents – Agent Communication Language).

Arquitectura del sistema para el desarrollo del sistema Multiagente.

La arquitectura que se usó para el desarrollo del sistema multiagente es la Arquitectura BDI, también se las conoce como Arquitectura deliberativas, ya que los agentes son iniciados y a partir de ese instante son capaces de deliberar, es decir, generar planes para alcanzar sus propios objetivos.

La arquitectura BDI (Belief, Desire, Intention) cuya traducción es creencias, deseos, intenciones es una de las arquitecturas más estudiadas y más usadas en la actualidad.

Para el desarrollo de la arquitectura del sistema multiagente para la comunicación de semáforos y trazar una ruta óptima en caso de que se produzca un accidente o

aglomeración se analizó y se estableció las creencias y deseos que un agente en este caso los semáforos pueda llegar a obtener.

A continuación las creencias y deseos que se encuentran en una Base de Datos.

Creencias.

- Hay comunicación.
- No hay comunicación.
- Hay accidente.
- No hay accidente.
- Nuevo Agente.
- No existe un nuevo Agente.
- Hay congestiónamiento.
- No hay congestiónamiento.

Deseos.

- Informar accidente.
- Informar congestiónamiento.
- Existe un nuevo agente.

Las intenciones es la manera en que los agentes responden ante eventos que perciben y tratan de alcanzar los objetivos planteados. En el sistema si un agente (semáforo) percibe que se ha producido un accidente o aglomeración, este semáforo se bloquea a sí mismo y a los 4 semáforos más próximos con el fin de evitar aglomeraciones y así poder trazar una ruta óptima para el conductor.

El envío de mensajes se realiza a través del estándar FIPA-ACL que es el protocolo de interacción para el intercambio de mensajes y actos de habla.

En la base de datos del sistema se encuentran también los semáforos que existen actualmente en la ciudad de Loja, características y posición geográfica, estos datos fueron brindados por el Municipio de Loja.

El sistema de este proyecto trabaja con agentes que pueden ser Agente emisor o receptor.

- Agente Emisor. Es el agente que informa que se ha producido un accidente o aglomeración a sus semáforos más cercanos.
- Agentes Receptor. Son agentes que reciben la información y al igual que el agente emisor estos se bloquean y se traza una ruta óptima para el conductor con la finalidad de evitar aglomeraciones.

1.3. Análisis y diseño de los agentes deliberativos.

1.3.1. Metodología de desarrollo.

Para el desarrollo del sistema multiagente se utiliza la metodología GAIA la cual es un enfoque práctico al análisis y diseño, implementación y gestión de sistemas de software basados en agentes inteligentes.

Las etapas básicas de la metodología GAIA son la fase de análisis y diseño.

1.3.2. Análisis.

En esta fase las principales tareas son definir una colección de roles de los agentes y definir las relaciones entre los agentes.

En esta parte del análisis del sistema multiagente se abordan dos aspectos: El modelo de rol y el modelo de interacción.

1.3.2.1. Modelo de roles.

El modelo de rol en el sistema multiagente define los roles del sistema, cada rol cada definido por las actividades, responsabilidades y protocolos asociados cada agente.

A continuación se define los roles del sistema, representados por los semáforos (Agentes).

Roles de los agentes:

- Establecer comunicación entre agentes.
- informar aglomeración entre agentes
- Informar accidentes.

TABLA I. ESTABLECER COMUNICACIÓN ENTRE AGENTES.

Esquema del Rol:	Establecer comunicación
Descripción.	Se encarga de establecer la comunicación entre los agentes existentes y nuevos agentes.
Actividades y protocolos.	. El nuevo agente envía mensajes a agentes cercanos. . Envío de mensajes entre agentes existentes. . Recepción de mensaje de disponibilidad del agente vecino.
Permisos:	
Lectura:	mensaje de disponibilidad.
Cambio:	
Genera:	respuesta si el agente está disponible.
Responsabilidades:	
Dinámicos:	
Seguridad.	

TABLA II. INFORMAR CONGESTIONAMIENTO ENTRE AGENTES.

Esquema del Rol:	Informar congestionamiento.
Descripción.	Se encarga de informar a los semáforos vecinos sobre un congestionamiento.
Actividades y protocolos.	Establecer comunicación con los agentes vecinos más cercanos. Enviar información que ha existido un congestionamiento. Confirmar recepción de mensaje de ubicación.
Permisos	Lectura: Mensaje de información de ubicación del accidente. Cambio: Semáforo vecino cambia de estado (Rojo) Genera:
Responsabilidades:	Dinámicos. Cambio el estado de los semáforos. Seguridad. Ninguno

TABLA III. INFORMAR ACCIDENTE.

Esquema del Rol	Informar accidente
Descripción.	Informar a los semáforos vecinos sobre un accidente.
Actividades y protocolos.	Establecer comunicación con los agentes vecinos más cercanos.

<p>Envía mensaje de información de accidente. Confirma recepción de mensaje de accidente.</p>
<p>Permisos: Lectura: mensaje de información de accidente Cambio: Genera:</p>
<p>Responsabilidades: Dinámicos. Seguridad.</p>

1.3.2.2. Modelo de Interacciones.

El modelo de interacciones define los protocolos a usar, el cual es el objetivo de la interacción y especificar los objetivos de emisor y receptor.

TABLA IV. ESTABLECER COMUNICACIÓN ENTRE AGENTES.

Protocolo	Establecer comunicación entre agentes.
Objetivo.	Se encarga de establecer la comunicación entre los agentes existentes y nuevos agentes.
Iniciador.	Agente emisor.
Receptor.	Agentes receptores.
Entradas.	Mensaje del agente emisor
Salidas.	Respuesta de los agentes receptores.
Procesamiento.	El agente iniciador envía un mensaje para saber si el agente se encuentra disponible, el receptor responde el mensaje para informar su disponibilidad.

TABLA V. INFORMAR CONGESTIONAMIENTO.

Protocolo	Informar congestionamiento.
Objetivo.	Se encarga de informar a los semáforos vecinos sobre un congestionamiento.
Iniciador.	Agente emisor.
Receptor.	Agentes receptores.
Entradas.	Los agentes receptores reciben información sobre un congestionamiento.
Salidas.	Agente cambia de estado bloqueando el acceso a los vehículos al lugar del congestionamiento.
Procesamiento.	El agente emisor, envía mensaje de información sobre congestionamiento a los agentes más cercanos, los agentes receptores envían mensaje de confirmación y cambian el estado del semáforo.

TABLA VI. INFORMAR ACCIDENTE.

Protocolo	Informar accidente
Objetivo.	Se encarga de informar a los semáforos vecinos sobre un accidente.
Iniciador.	Agente emisor.

Receptor.	Agentes receptor.
Entradas.	Los agentes receptores reciben información sobre un accidente.
Salidas.	Agente cambia de estado bloqueando el acceso a los vehículos al lugar del accidente.
Procesamiento.	El agente emisor, envía mensaje de información sobre accidente a los agentes más cercanos, los agentes receptores envían mensaje de confirmación y cambian el estado del semáforo.

1.3.3. Diseño del sistema

Es la fase concreta del diseño del sistema, en el modelo de agentes se define los agentes que existen y qué papel juega en cada agentes y el modelo de comunicaciones representa el enlace entre los agentes y el proceso de comunicación de los mismos.

1.3.3.1. Modelo de agentes.

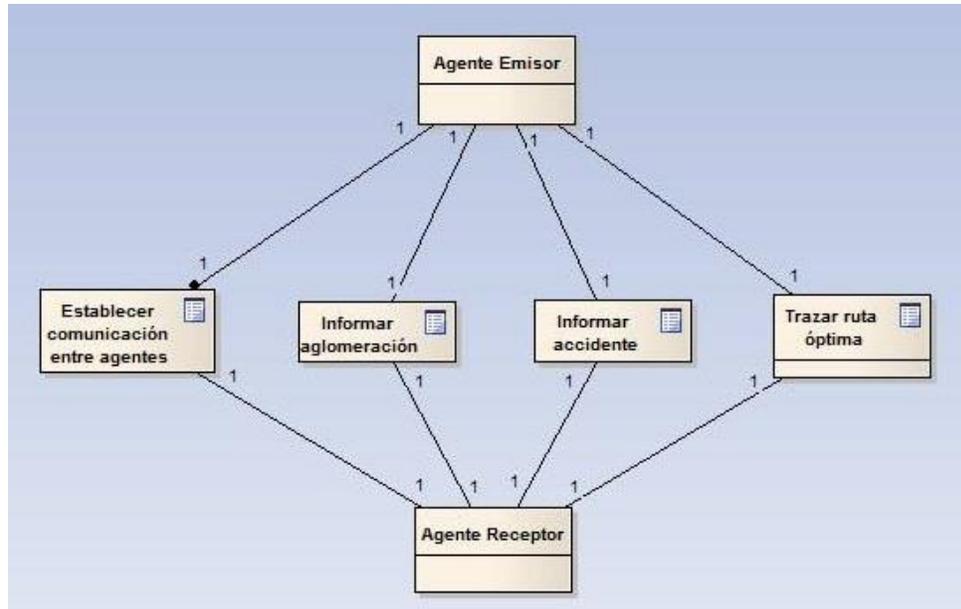


FIGURA 10. Modelo de Agentes.

1.3.3.2. Modelo de comunicaciones.

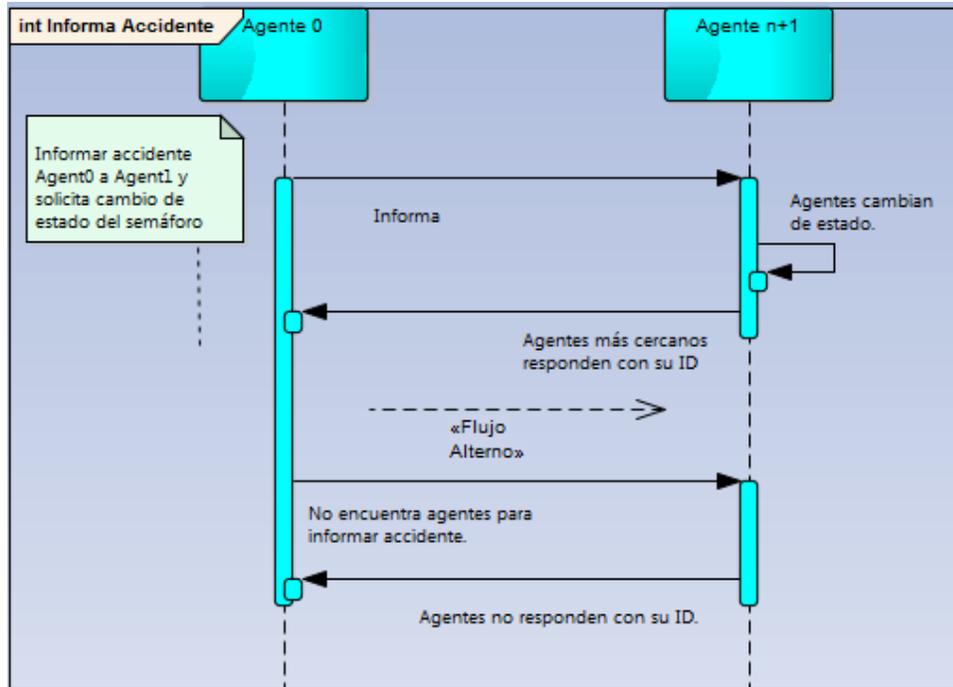


FIGURA 11. Establecer comunicación entre agentes.

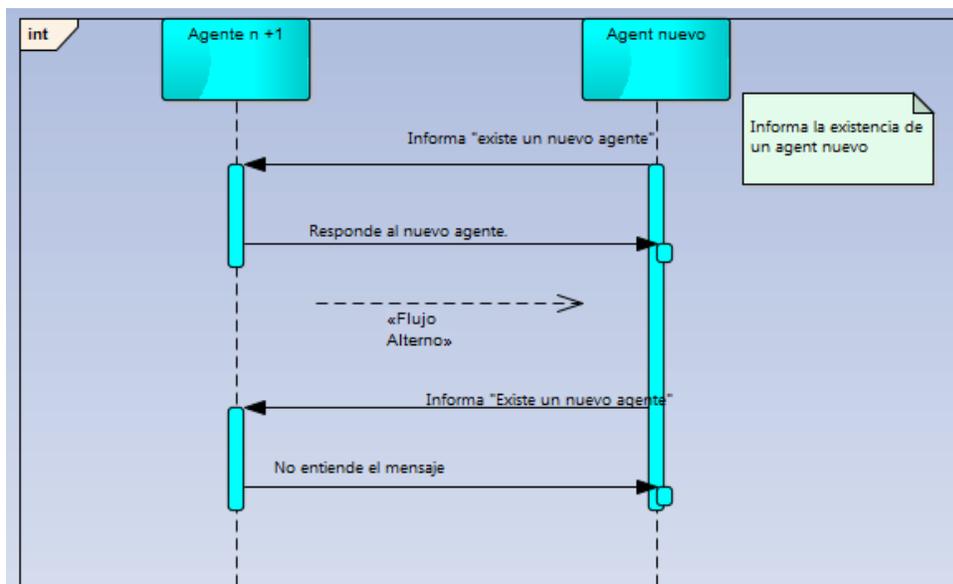


FIGURA 12. Informar Nuevo Agente.

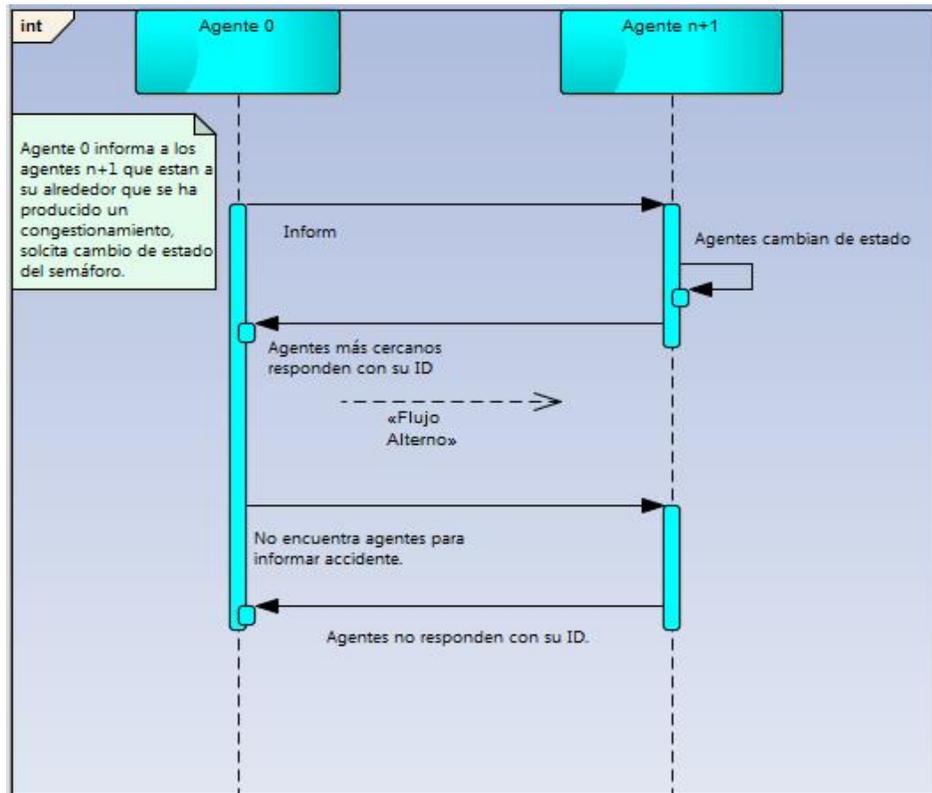


FIGURA 13. Informar congestionamiento.

1.4. Programar los agentes para la comunicación de semáforos

El lenguaje que se utilizó fue Java, para la creación de las vistas se utilizó la librería Primefaces la cual facilita la creación de aplicaciones web, para la creación, manejo y comunicación de agentes se utiliza la librería jadeBin, la cual contiene un conjunto de archivos .jar y que se integra con el estándar FIPA-ACL para la comunicación.

1.4.1. Inicio del contenedor principal.

El contenedor principal de JADE es donde se almacenan los agentes y donde inician, para ello se creó una clase denominada ContenedorAgentes.java, se importa de la librería Jade la clase AgenteContainer.java que se utiliza para agregar o insertar un contenedor en la plataforma Jade.

```

Información: -----
This is JADE 4.3.1 - revision 6695 of 2013/12/03 14:41:54
downloaded in Open Source, under LGPL restrictions,
at http://jade.tilab.com/
-----
Información: Listening for intra-platform commands on address:
- jicp://10.10.10.109:1099

Información: Service jade.core.management.AgentManagement initialized
Información: Service jade.core.messaging.Messaging initialized
Información: Service jade.core.resource.ResourceManagement initialized
Información: Service jade.core.mobility.AgentMobility initialized
Información: Service jade.core.event.Notification initialized
Información: HTTP-MTP Using XML parser com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser
Información: MTP addresses:
http://Usuario-PC:7778/acc
Información: -----
Agent container Main-Container@10.10.10.109 is ready.
-----

```

FIGURA 14. Contenedor principal de Jade.

El contenedor principal está programado para que inicie cuando el usuario o administrador ingrese al sistema, una vez que arranca el contenedor, los semáforos inician y están en espera de ser activados.

```

Información: MTP addresses:
http://Usuario-PC:7778/acc
Información: -----
Agent container Main-Container@10.10.10.109 is ready
-----
Información: Hibernate: select agentes0_age_id as
Información: Iniciando los agentessemáforo-001
Información: Iniciando los agentessemáforo-002
Información: Iniciando los agentessemáforo-003
Información: Iniciando los agentessemáforo-004
Información: Iniciando los agentessemáforo-005
Información: Iniciando los agentessemáforo-006
Información: Iniciando los agentessemáforo-007
Información: Iniciando los agentessemáforo-008

```

FIGURA 15. Inicio de agentes.

Todos los semáforos que están almacenados en la Base de datos serán emisores y receptores, un agente emisor informará a sus más cercanos que en su intersección se ha producido un accidente, los semáforos receptores recibirán dicho mensaje y viceversa.

Para crear los emisores y receptores se creó una clase en Java denominada EmisorReceptor.java la cual activa los agentes, una vez que los agentes se activan hay que añadirles los comportamientos necesarios para que cumplan sus tareas o objetivos, eso a través de un método denominado doActivate() creado en la misma clase EmisorReceptor.java

```
public class EmisorReceptor extends Agent {

    private AgenteDao agenteDao;

    @Override
    public void doActivate() {
        super.doActivate();
        //To change body of generated methods, choose Tools | Templates.
        agenteDao = new AgenteDaoImpl();
        Agente agente = agenteDao.findByCodigo(this.getLocalName());
        if (agente.getAgeEstado() != null) {
            System.out.println("consulata el estado es correcto");
            ComportamientoEmisor comportamientoEmisor = new ComportamientoEmisor(agente);
            comportamientoEmisor.setAgent(this);
            addBehaviour(comportamientoEmisor);
        } else {
            ComportamientoReceptor comportamientoSimple = new ComportamientoReceptor();
            comportamientoSimple.setAgent(this);
            addBehaviour(comportamientoSimple);
        }
    }
}
```

FIGURA 16. Clase EmisorReceptor.

Los comportamientos definen la acción a ser ejecutada, hacen referencia a una funcionalidad específica que incorpora un agente, dichos comportamientos especifican los servicios que realiza un agente para lograr sus objetivos.

Los objetivos del comportamiento emisor será el de informar a los receptores que se ha producido un accidente, aglomeración o que se ha incorporado un nuevo agente, el agente emisor logra estos objetivos a través de la creencias y deseos almacenados en la Base de datos, el comportamiento es invocado cuando se produce un evento asociado al comportamiento, por ejemplo cuando se produce un accidente.

Los objetivos del comportamiento receptor es el de recibir el mensaje, procesar el mensaje e informar a sus emisor que el mensaje ha sido recibido con su identificación y su nombre.

Para el envío de mensaje se utiliza el FIPA-ACL que es el estándar de comunicación para agentes, FIPA-ACL provee los protocolos necesarios para que el mensaje enviado llegue en correcto estado.

La comunicación entre agentes es fundamental, con ello se obtiene la potencia propia de los sistemas multiagente.

Para que el agente emisor pueda enviar mensajes se crea un método en la clase ComportamientoEmisor.java llamado ACLMensaje y se sigue los siguientes pasos.

- Se crea un objeto ACLMensaje, que se extiende de la clase principal de jade.lang.acl.ACLMensaje;
- Se utiliza el método ACLMensaje para rellenar los campos necesarios.
- Se invoca al método send() de la clase Agent.

En la siguiente imagen se ilustra el envío de mensajes programado en java para el emisor y que envía a sus semáforos más cercanos.

```
private void ACLMensaje(Agentes agentes, String mensajeC) {
    List<Agentes> lista = agenteDao.obtenerAgentesCompleto(agentes.getAgeLatitud(), agentes.getAgeLongitud());
    lista.remove(0);
    ACLMensaje mensaje = new ACLMensaje(ACLMensaje.INFORM);
    for (Agentes agentes1 : lista) {
        System.out.println("mensajes a enviar mensajes" + agentes1.getAgeCodigo());
        mensaje.setSender(getAgent().getAID()); //quién envía el mensaje
        AID id = new AID();
        id.setLocalName(agentes1.getAgeCodigo());
        mensaje.addReceiver(id);
    }
    mensaje.setLanguage("Español"); //idioma del mensaje
    //quien recibe el mensaje
    mensaje.setContent(mensajeC); //contenido del mensaje
    getAgent().send(mensaje); //obtener el agente para enviar mensaje
    ACLMensaje msj2 = getAgent().blockingReceive();
    System.out.println("El agente " + msj2.getSender().getLocalName()
        + " dice: " + msj2.getContent() + " y el Objeto es: ");
    if (msj2 != null) {
        getAgent().doSuspend();
    }
}
```

FIGURA 17. Código Java envió mensajes semáforo más cercano.

Para recibir los mensajes y responder se usa el método de la clase Agent, llamado blockingReceive (), ya que este bloquea instantáneamente todos los comportamientos del agente hasta que llegue un mensaje, además devuelve el mensaje recibido que liberó el bloqueo.

En la siguiente figura se observa cómo se programó en Java la recepción de mensajes y la respuesta.

```
ACLMessage mensaje = getAgent().blockingReceive();
System.out.println("mensaje:"+mensaje);

if (mensaje != null) {

    System.out.println("Mensaje recibido");
    if (mensaje.getContent().equalsIgnoreCase("Informar accidente")) {
        System.out.println("El agente " + mensaje.getSender().getLocalName()
            + " dice: " + mensaje.getContent());
        System.out.println("Agente:" +getAgent().getLocalName()+" responde");
        ACLMessage mensaje2 = new ACLMessage(ACLMessage.INFORM);
        mensaje2.setSender(getAgent().getAID());
        mensaje2.addReceiver(mensaje.getSender());
        mensaje2.setLanguage("Español");
        mensaje2.setContent("deliberando");
        getAgent().send(mensaje2);
        System.out.println("Informar accidente");
    }
}
```

FIGURA 18. Código Java Recepción de mensajes.

Una vez que los agentes están programados para poder enviar mensajes con sus respectivos comportamientos y los receptores para recibirlos y responder, se puede simular un accidente, aglomeración y agregar un nuevo semáforo para observar cómo se produce la comunicación entre ellos.

```

-----
Información: El agente 067 dice: Informar accidente
Información: Agente:068 responde
Información: mensaje:(INFORM
:sender ( agent-identifier :name "067@192.168.0.105:1099/JADE"
:addresses (sequence http://Equipo:61881/acc ))
:receiver (set ( agent-identifier :name "066@192.168.0.105:1099/JADE" )
( agent-identifier :name "069@192.168.0.105:1099/JADE" )
( agent-identifier :name "068@192.168.0.105:1099/JADE" )
( agent-identifier :name "055@192.168.0.105:1099/JADE" ) )
:content "Informar accidente"
:language Español )
Información: Mensaje recibido
-----

Información: El agente 067 dice: Informar accidente
Información: Agente:055 responde
Información: mensaje:(INFORM
:sender ( agent-identifier :name "067@192.168.0.105:1099/JADE"
:addresses (sequence http://Equipo:61881/acc ))
:receiver (set ( agent-identifier :name "066@192.168.0.105:1099/JADE" )
( agent-identifier :name "069@192.168.0.105:1099/JADE" )
( agent-identifier :name "068@192.168.0.105:1099/JADE" )
( agent-identifier :name "055@192.168.0.105:1099/JADE" )
)
:content "Informar accidente"
:language Español )
Información: Mensaje recibido

```

FIGURA 19. Mensaje para informar un accidente.

```

Información: mensajes a enviar mensjes129
Información: mensajes a enviar mensjes042
Información: mensajes a enviar mensjes054
Información: mensajes a enviar mensjes065
Información: mensaje:(INFORM
:sender ( agent-identifier :name "053@192.168.0.105:1099/JADE"
:addresses (sequence http://Equipo:63344/acc ))
:receiver (set ( agent-identifier :name "129@192.168.0.105:1099/JADE" )
( agent-identifier :name "042@192.168.0.105:1099/JADE" )
(agent-identifier :name "054@192.168.0.105:1099/JADE" )
(agent-identifier :name "065@192.168.0.105:1099/JADE" ) )
:content "Informar congestionamiento"
:language Español )
Información: Mensaje recibido
-----
Información: El agente 053 dice: Informar congestionamiento
Información: Agente:054 responde
Información: mensaje:(INFORM
:sender ( agent-identifier :name "053@192.168.0.105:1099/JADE"
:addresses (sequence http://Equipo:63344/acc ))
:receiver (set ( agent-identifier :name "129@192.168.0.105:1099/JADE" )
( agent-identifier :name "042@192.168.0.105:1099/JADE" )
(agent-identifier :name "054@192.168.0.105:1099/JADE" )
(agent-identifier :name "065@192.168.0.105:1099/JADE" ) )
:content "Informar congestionamiento"
:language Español )
Información: Mensaje recibido
-----

```

FIGURA 20. Mensaje para informar un congestionamiento.

1.5. Permitir la creación e integración de los nuevos agentes deliberativos.

Gracias a Jade, es posible crear agentes de una manera más simple y hacer que se integren de manera simultánea y que se comunique a sus vecinos más cercanos informándoles de que es nuevo.

Para crear un nuevo agentes se creó un método denominado agregarMarcadores(), la cual realiza un recorrido a la base de datos para poder almacenar su nombre y sus coordenadas.

```

private void agregarMarcadores() {
    List<Agentes> listaAgentes = agenteDao.findAll();
    LatLng coordenada;
    for (Agentes agentes : listaAgentes) {
        coordenada = new LatLng(agentes.getAgeLatitud(), agentes.getAgeLongitud());
        emptyModel.addOverlay(new Marker(coordenada, agentes.getAgeCodigo(), null, "http://www.kitlatren");
    }
}

```

FIGURA 21. Método agregarMarcadores.

En la vista del mapa de la ciudad de Loja

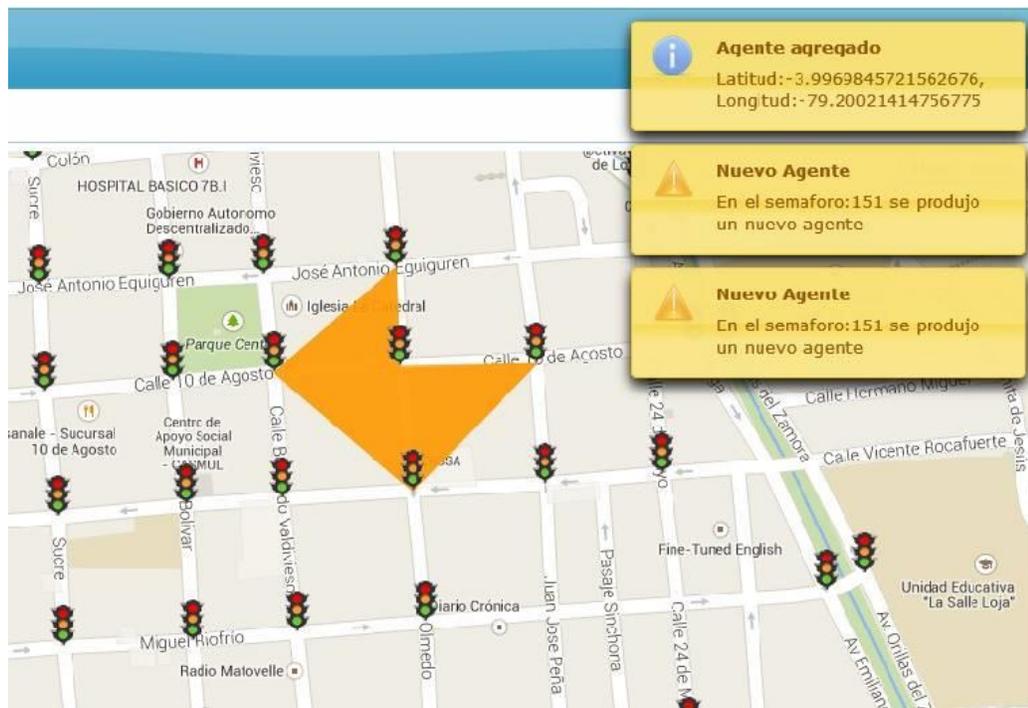


FIGURA 22. Nuevo agente agregado.

Los mensajes son de la siguiente manera.

```

Información: mensajes a enviar mensjes044
Información: mensajes a enviar mensjes056
Información: mensajes a enviar mensjes150
Información: El agente 151 dice: existe nuevo agente
Información: El agente 038 dice: deliberando y el Objeto es:
Información: mensaje:(INFORM
:sender ( agent-identifier :name "151@192.168.0.105:1099/JADE"
:addresses (sequence http://Equipo:64068/acc ))
:receiver (set ( agent-identifier :name "038@192.168.0.105:1099/JADE" )
( agent-identifier :name "044@192.168.0.105:1099/JADE" )
(agent-identifier :name "056@192.168.0.105:1099/JADE" )
(agent-identifier :name "150@192.168.0.105:1099/JADE" ) )
:content "existe nuevo agente"
:language Español )
Información: Agente:038 responde
Información: Informar nuevo agente
Información: Mensaje recibido
-----
Información: El agente 151 dice: existe nuevo agente
Información: mensaje:(INFORM
:sender ( agent-identifier :name "151@192.168.0.105:1099/JADE"
:addresses (sequence http://Equipo:64068/acc ))
:receiver (set ( agent-identifier :name "038@192.168.0.105:1099/JADE" )
( agent-identifier :name "044@192.168.0.105:1099/JADE" )
(agent-identifier :name "056@192.168.0.105:1099/JADE" )
(agent-identifier :name "150@192.168.0.105:1099/JADE" ) )
:content "existe nuevo agente"
:language Español )
Información: Agente:044 responde
Información: Informar nuevo agente

```

FIGURA 23. Mensajes cuando se agrega un nuevo agente.

2. SEGUNDA FASE: Desarrollar algoritmos que permitan establecer rutas óptimas mediante la ubicación de semáforos.

2.1. Analizar que algoritmo se utilizará para la creación de los grafos.

Para encontrar rutas mínimas es necesario calcular el camino más corto desde un nodo de origen a un nodo de destino. Existen dos algoritmos los cuales nos hemos basado para diseñar nuestro propio algoritmo que evite nodos que intervengan en algún evento por parte de los agentes ya sea congestión o accidente que permitan trazar la ruta óptima los algoritmos son el de Dijkstra y el A-star (A*).

Para nuestro análisis debemos tomar en cuenta si un nodo interviene en un evento producido por algún agente ya sea accidente y congestionamiento. El algoritmo busca nodos q se encuentran cercanos al nodo inicial o nodo padre, tiene que evitar a ciertos nodos que se encuentran en el mapa ya que perteneces a los agentes que no están disponibles y que intervienen en algún evento ya sea accidente o congestionamiento.

2.2. Diseñar grafos que permitan trazar la ruta óptima.

2.2.1. Diseño de grafos.

En el diseño del grafo, luego del análisis correspondiente se llegó a la conclusión de que la mejor manera para poder trazar la ruta evitando ciertos nodos o coordenadas en el mapa, que corresponden a los agentes que no están disponibles o sea donde existe algún congestionamiento o accidente; es almacenando una lista con coordenadas por donde queremos que pase la ruta, ya que los mapas de google permiten ingresar varias coordenadas o hitos por donde se quiera trazar la ruta, con lo cual se evitaría las coordenadas de los agentes que están inhabilitados.

2.2.2. Codificación.

A continuación mostramos los principales métodos que se desarrollaron para la codificación de la ruta mediante el algoritmo establecido.

Tanto para el Método para determinar el agente más cercano al usuario y el método que permite determinar el agente más cercano al destino del usuario se utilizó la siguiente formula que a continuación describiremos:

```
Formula simple: 6378.137 * Acos(x)
6378.137 Diametro de la tierra en KM: wgs86
Acos(X): Arco coseno de X
donde X = Cos( Lat1 ) * Cos( Lat2 ) * Cos( Lon2 - Lon1 ) + Sin(
Lat1 ) * Sin( Lat2 )
```

Método para determinar el agente más cercano al usuario

```
public Agentes agenteMasCercano(LatLng coordenadaUsuario) {
    Double distancia;
    List<Agentes> listaAgentes = agenteDao.findAll();
    Agentes agenteAux = new Agentes();
    Double distanciaAux = 6378.137 * Math.acos(Math.cos(coordenadaUsuario.getLat())
        * Math.cos(listaAgentes.get(0).getAgeLatitud()) * Math.cos(listaAgentes.get(0).getAgeLongitud()
        - coordenadaUsuario.getLng() + Math.sin(coordenadaUsuario.getLat())
        * Math.sin(listaAgentes.get(0).getAgeLatitud()));
    for (Agentes agente : listaAgentes) {
        distancia = 6378.137 * Math.acos(Math.cos(coordenadaUsuario.getLat()) * Math.cos(agente.getAgeLatitud())
            * Math.cos(agente.getAgeLongitud() - coordenadaUsuario.getLng() + Math.sin(coordenadaUsuario.getLat())
            * Math.sin(agente.getAgeLatitud()));
        if (distancia < distanciaAux) {
            distanciaAux = distancia;
            agenteAux = agente;
        }
    }
    if (agenteAux.getAgeEstado() != null) {
        return null;
    } else {
        return agenteAux;
    }
}
```

FIGURA 24. Código agente más cercano al usuario.

Método que permite determinar el agente más cercano al destino del usuario.

```
return
*/
public Agentes masCercanoAlDestino(Agentes agenteHijo1, Agentes agenteHijo2, LatLng coordenadaUsuarioFinal) {
    Double distancia1 = 6378.137 * Math.acos(Math.cos(coordenadaUsuarioFinal.getLat())
        * Math.cos(agenteHijo1.getAgeLatitud()) * Math.cos(agenteHijo1.getAgeLongitud()
        - coordenadaUsuarioFinal.getLng() + Math.sin(coordenadaUsuarioFinal.getLat())
        * Math.sin(agenteHijo1.getAgeLatitud()));
    Double distancia2 = 6378.137 * Math.acos(Math.cos(coordenadaUsuarioFinal.getLat())
        * Math.cos(agenteHijo2.getAgeLatitud()) * Math.cos(agenteHijo2.getAgeLongitud()
        - coordenadaUsuarioFinal.getLng() + Math.sin(coordenadaUsuarioFinal.getLat())
        * Math.sin(agenteHijo2.getAgeLatitud()));
    if (distancia1 < distancia2 && agenteHijo1.getAgeEstado() == null) {
        return agenteHijo1;
    } else if (distancia2 < distancia1 && agenteHijo2.getAgeEstado() == null) {
        return agenteHijo2;
    } else if (agenteHijo1.getAgeEstado() != null && agenteHijo2.getAgeEstado() == null) {
        return agenteHijo2;
    } else if (agenteHijo2.getAgeEstado() != null && agenteHijo1.getAgeEstado() == null) {
        return agenteHijo1;
    } else {
        return null;
    }
}
/**
```

FIGURA 25. Código agente más cercano al destino del usuario.

Método para determinar las coordenadas de los puntos por los cuales se debe reconfigurar la ruta del usuario, tomando como inicio su coordenada inicial y como fin su coordenada final.

```
public ArrayList<LatLng> listaHitos(LatLng coordenadaUsuarioInicial, LatLng coordenadaUsuarioFinal) {
    System.out.println("Si ingreso");
    Agentes agentePadreInicial = agenteMasCercano(coordenadaUsuarioInicial);
    Agentes agenteHijo;
    ArrayList<LatLng> listaCoordenadass = new ArrayList<>();
    boolean terminar = Boolean.TRUE;
    LatLng coordenadaGrafo = new LatLng(agentePadreInicial.getAgeLatitud(), agentePadreInicial.getAgeLongitud());
    listaCoordenadass.add(coordenadaGrafo);
    do {
        agenteHijo = agenteHijoMasCercano(agentePadreInicial);
        if (agenteHijo != null) {
            coordenadaGrafo = new LatLng(agenteHijo.getAgeLatitud(), agenteHijo.getAgeLongitud());
        }
        if (agenteHijo != null && listaCoordenadass.size()<8) {
            if (!agenteHijo.getAgeCodigo().equals(agentePadreInicial.getAgeCodigo())) {
                listaCoordenadass.add(coordenadaGrafo);
                agentePadreInicial = agenteHijo;
            }
        } else {
            terminar = Boolean.FALSE;
        }
    } while (terminar);
    if(listaCoordenadass.size()==1)
        listaCoordenadass.remove(listaCoordenadass.size() - 1);
    return listaCoordenadass;
}
```

FIGURA 26. Reconfigurar ruta.

2.3. Desarrollar el algoritmo que permita trazar una ruta óptima ante congestionamiento vehicular o accidentes.

El diseño del algoritmo para encontrar las coordenadas del grafo es el siguiente:

1. Inicio
2. Ingresa la coordenada inicial
3. Ingresa la coordenada final
4. Se crea una lista con todas las coordenadas correspondientes a los nodos o agentes.
5. Se crea una lista que almacenará las coordenadas del grafo (por donde se debe trazar la ruta.)
6. Se busca el agente más cercano a la coordenada inicial, tomará el nombre de agente padre.
7. Se busca el agente más cercano a la coordenada final.

8. Se busca el agente hijo más cercano del agente padre y que esté disponible (que no exista congestión ni accidente), si existe pasa a ser padre y se asigna esa coordenada a la lista de coordenadas del grafo y se sigue buscando más agentes hijos hasta llegar al agente más cercano a la coordenada final.
9. En caso de que existan puntos almacenados en la lista de coordenadas del grafo, se traza la ruta desde la coordenada inicial hasta la coordenada final, pasando por las coordenadas del grafo.
10. En caso que no exista un nodo hijo disponible, la ruta se traza normalmente sin ninguna coordenada en el grafo, tal como se traza en los mapas de google.
11. Fin

En la imagen siguiente se puede ver la forma en la que funciona el algoritmo.

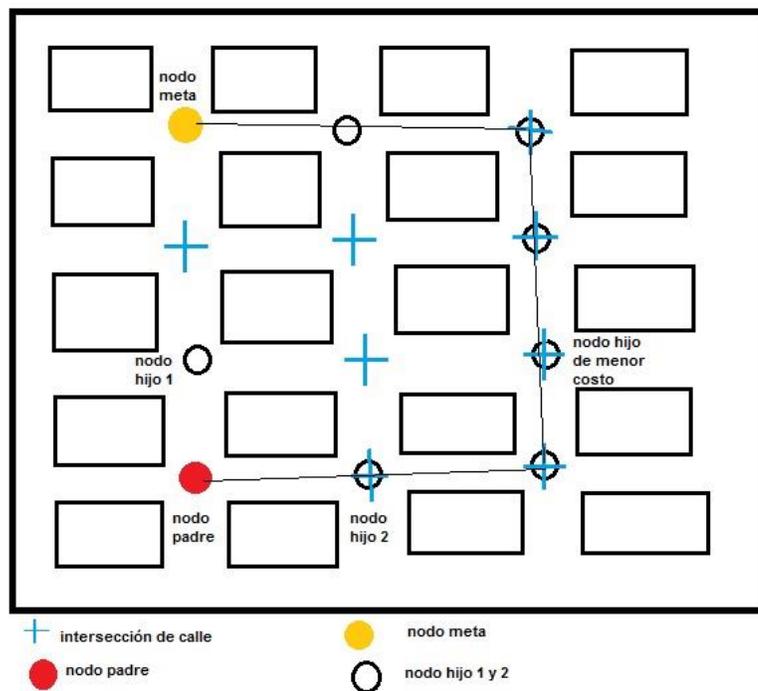


FIGURA 27. Funcionamiento grafo.

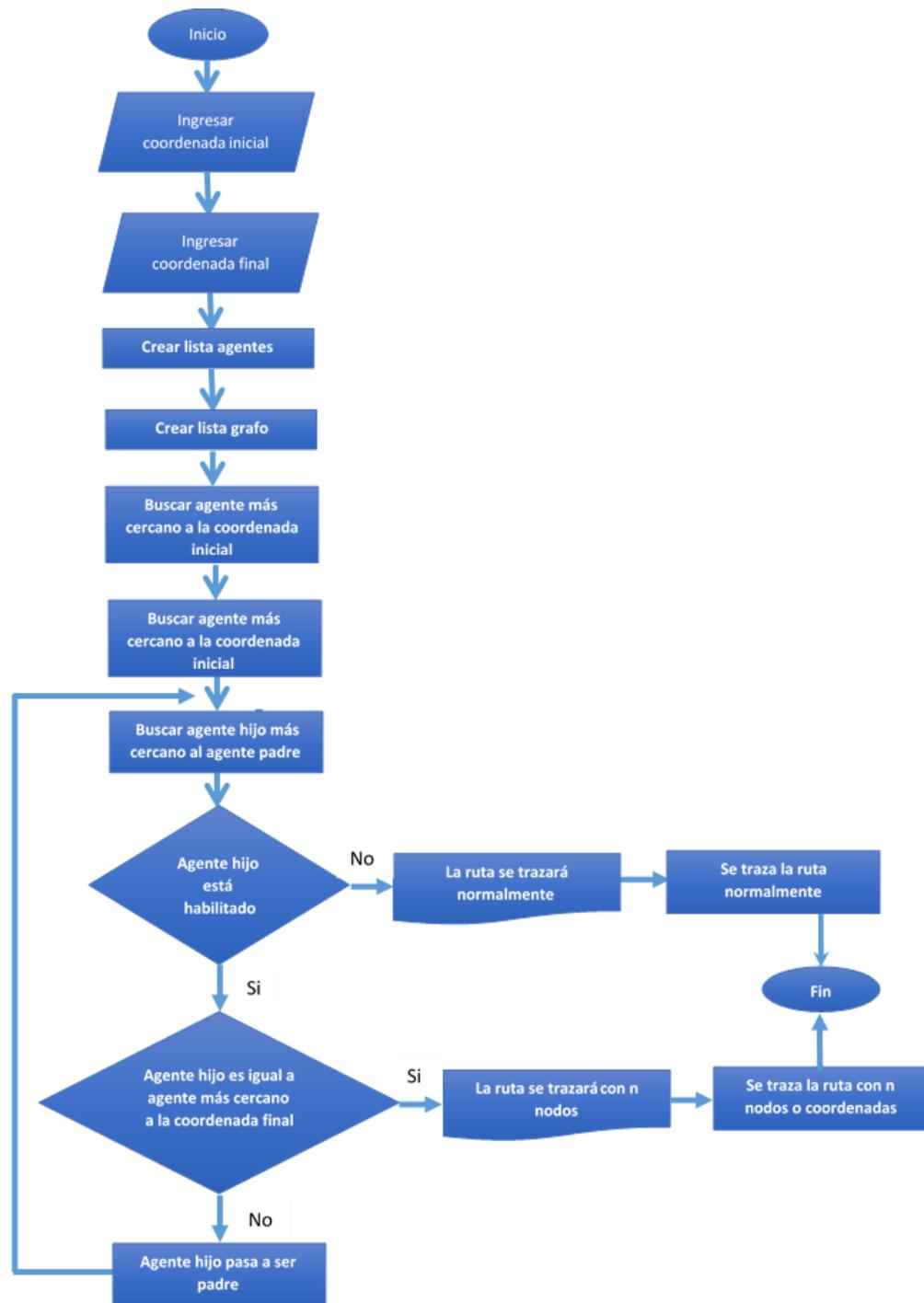


FIGURA 28. Diseño del Algoritmo.

2.4. Realizar las pruebas de trazado de ruta, se utiliza el mapa en base a la ubicación de los semáforos.

Prueba 1.

Accidente: Calle 10 de Agosto y Sucre

Congestionamiento: V. Rocafuerte y Bolívar.

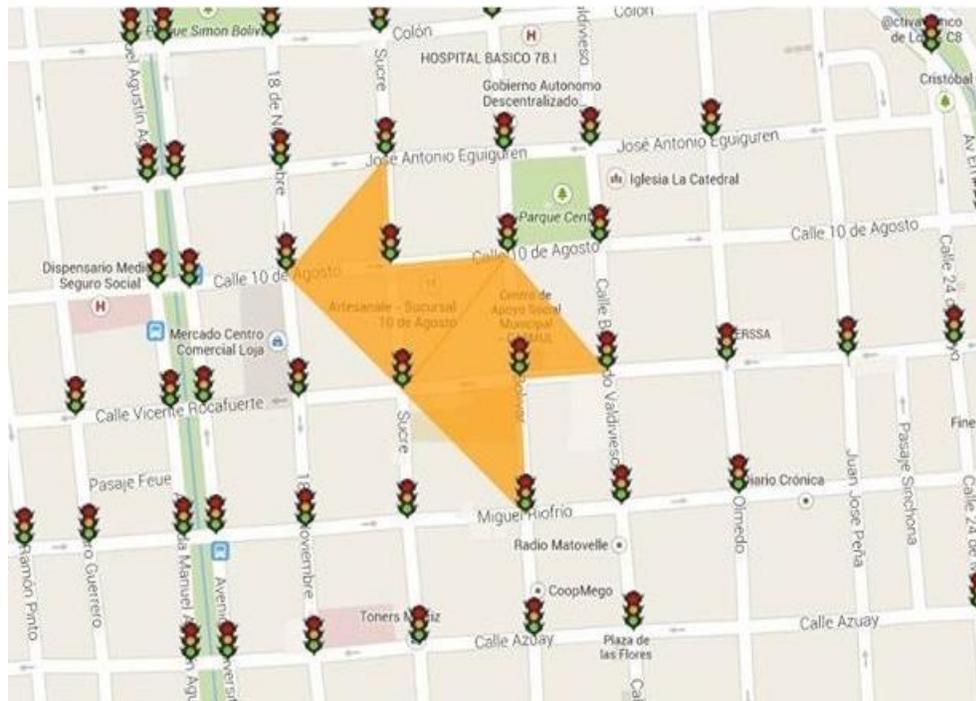


FIGURA 29. Ubicación eventos.

Lugar de origen A: José Antonio Eguiguren y 18 de noviembre

Lugar de destino B: Juan José Peña y Vicente Rocafuerte.



FIGURA 30. Resultado prueba.

Prueba 2.

Accidentes: 18 de Noviembre y V. Rocafuerte, Bolívar y V. Rocafuerte, Bernardo Valdivieso y Miguel Riofrío, Sucre y Miguel Riofrío.

Aglomeraciones: Bolívar y Miguel Riofrío, Bolívar y Azuay.

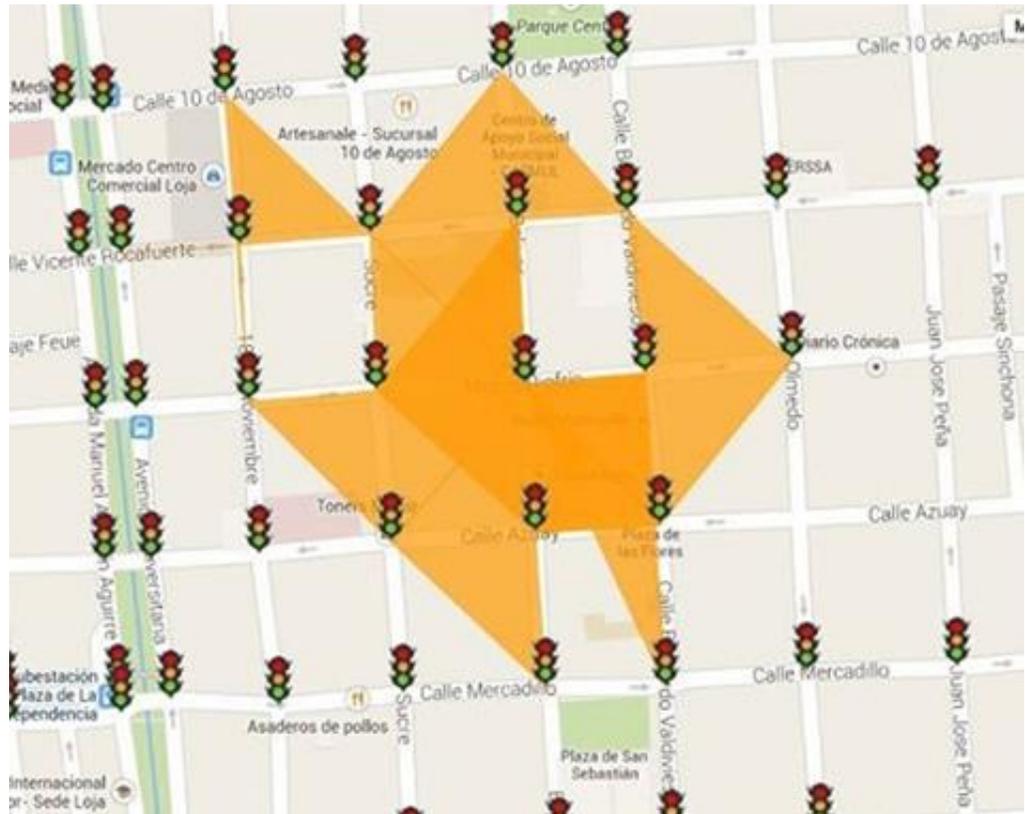


FIGURA 31. Ubicación eventos 2.

Lugar de origen: 10 de agosto y Bolívar

Lugar de destino: Calle Olmedo y Lourdes

3. TERCERA FASE: Integrar y simular agentes inteligentes y algoritmo de rutas óptimas para evitar aglomeraciones y accidentes.

3.1. Diseño de la interfaz del sistema.

Se realiza una interfaz principal del sistema, la cual permitirá al usuario poder agregar semáforos al mapa de forma simple, simular accidentes o aglomeraciones y simular ruta a un destino dado estos por el usuario.

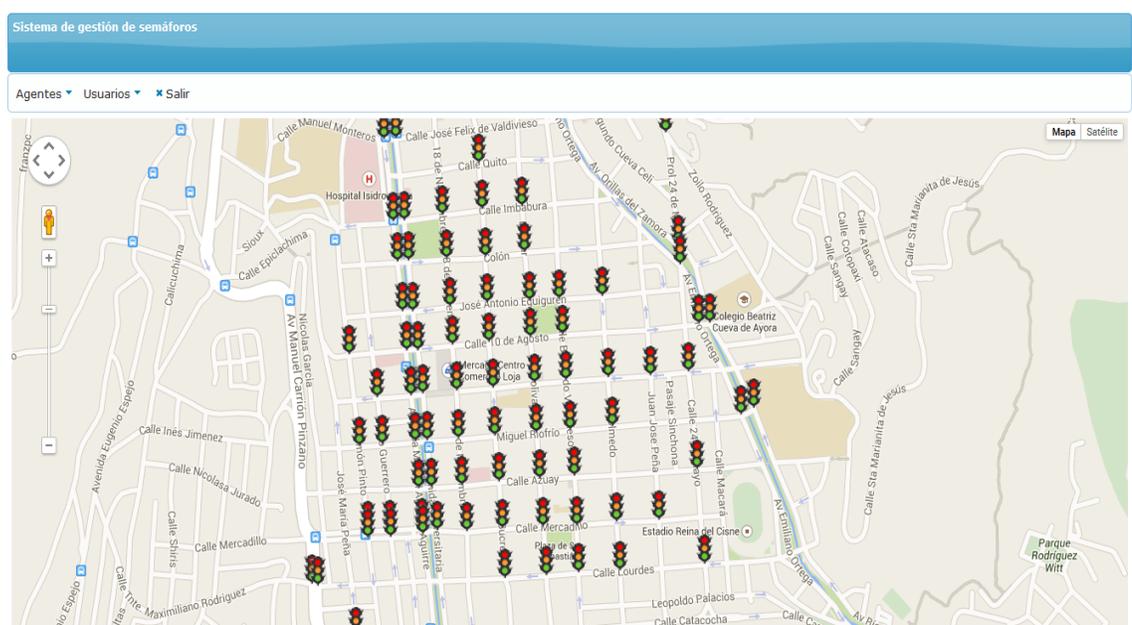


FIGURA 33. Ventana Principal del Sistema.

Al momento de crear los semáforos, el usuario dará click en la intersección correspondiente para poder crear los semáforos la cual ingresará las características correspondientes a los semáforos.

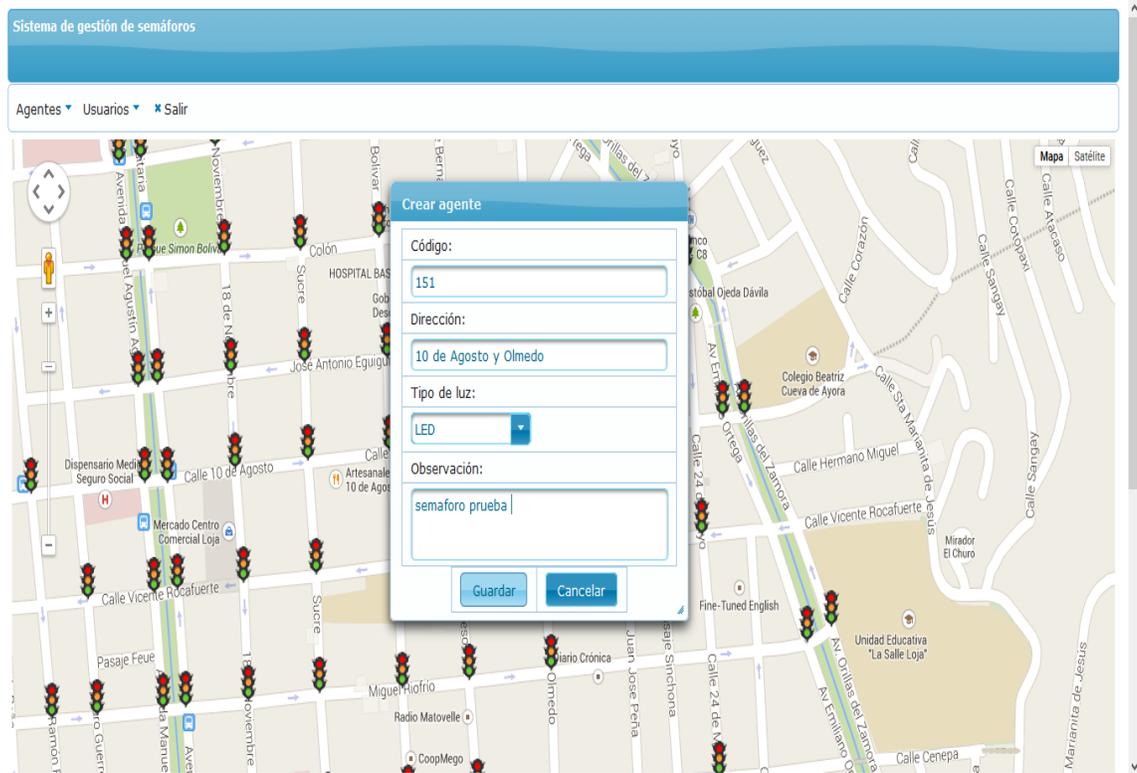


FIGURA 34. Crear Agente.

Cada vez que simulara un accidente o congestionamiento se diseñó una ventana, que permitirá elegir entre accidente o congestionamiento y la cual se producirá el accidente e internamente se producirá el envío de mensajes.

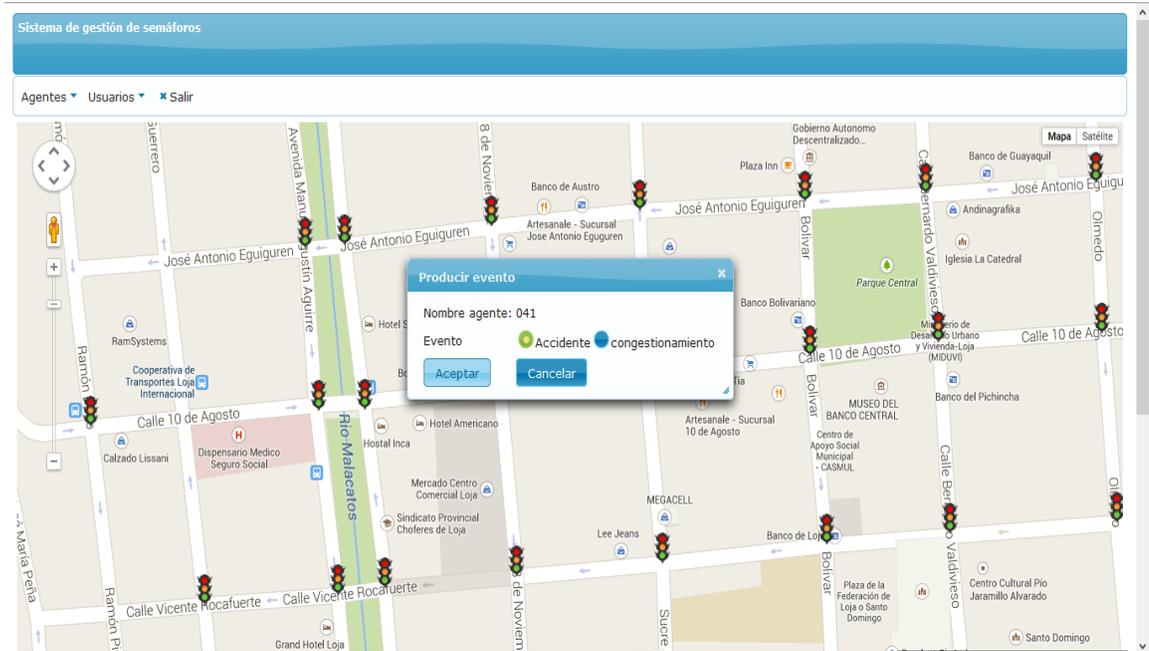


FIGURA 35. Producir Evento (Accidente o Congestionamiento).

Cuando se produce el accidente o aglomeración se marcará en el mapa el lugar exacto donde se produjo los eventos, conociendo los lugares donde existe accidente o aglomeración el usuario podrá elegir el lugar de origen donde se encuentre y el lugar de destino y se trazara una ruta evitando los eventos producidos.

Prueba 1.

Accidente: 10 de Agosto y 18 de noviembre, agente 041.

Lugar de origen A: 18 de noviembre y José Antonio Eguiguren.

Lugar de destino B: 18 de noviembre y Miguel Riofrío.

Si evito el accidente.

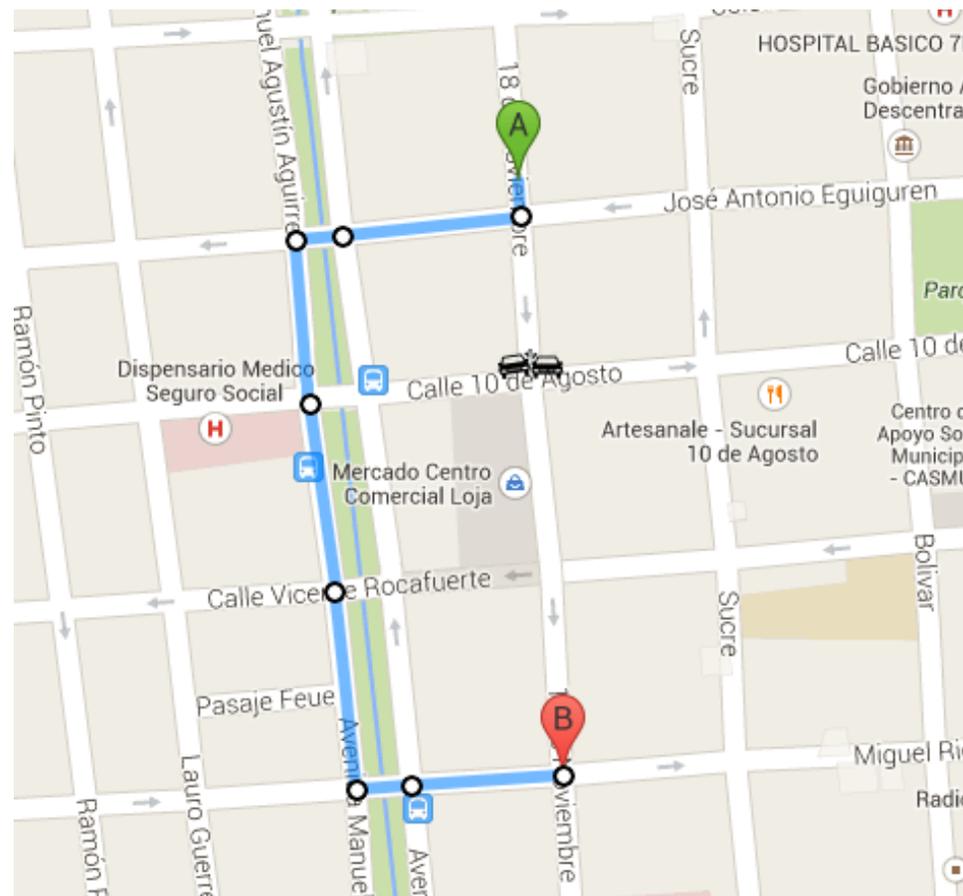


FIGURA 37. Prueba 1.

Prueba 2.

Congestionamiento: 10 de Agosto y Bolívar, agente 043.

Lugar de origen A: Bolívar y Miguel Riofrío.

Lugar de destino B: Bolívar y Colón.

Si evita el Congestionamiento.

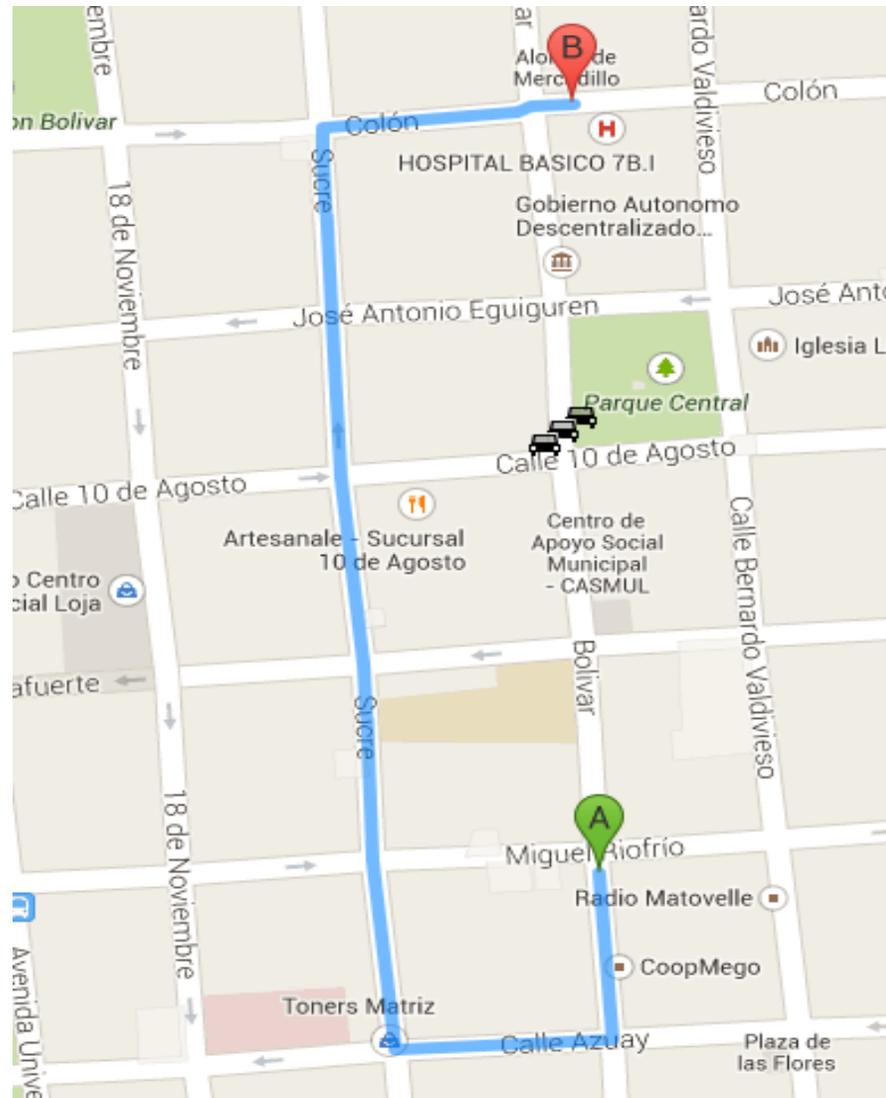


FIGURA 38. Prueba 2.

Prueba 3.

Accidente: Miguel Riofrío y Sucre, agente 065.

Lugar de origen A: Miguel Riofrío y 18 de noviembre.

Lugar de destino B: Miguel Riofrío y Juan José Peña.

Si evita el Accidente.

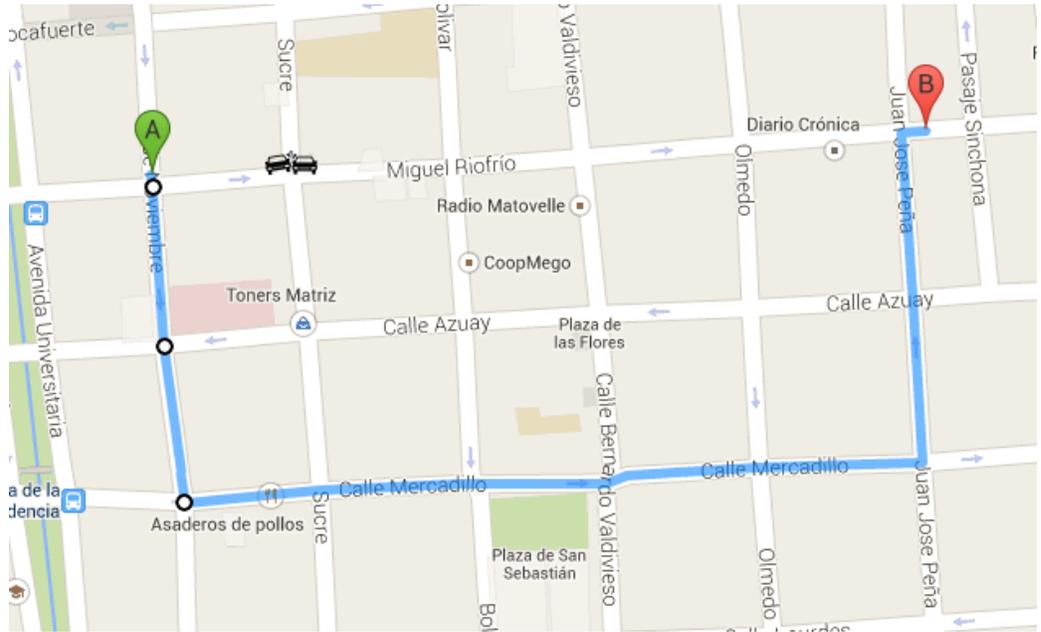


FIGURA 39. Prueba 3.

Prueba 4.

Congestionamiento: Rocafuerte y Bolívar, agente 054.

Lugar de origen A: Rocafuerte y Bernardo Valdivieso.

Lugar de destino B: Rocafuerte y 18 de Noviembre.

Si evita el Congestionamiento.

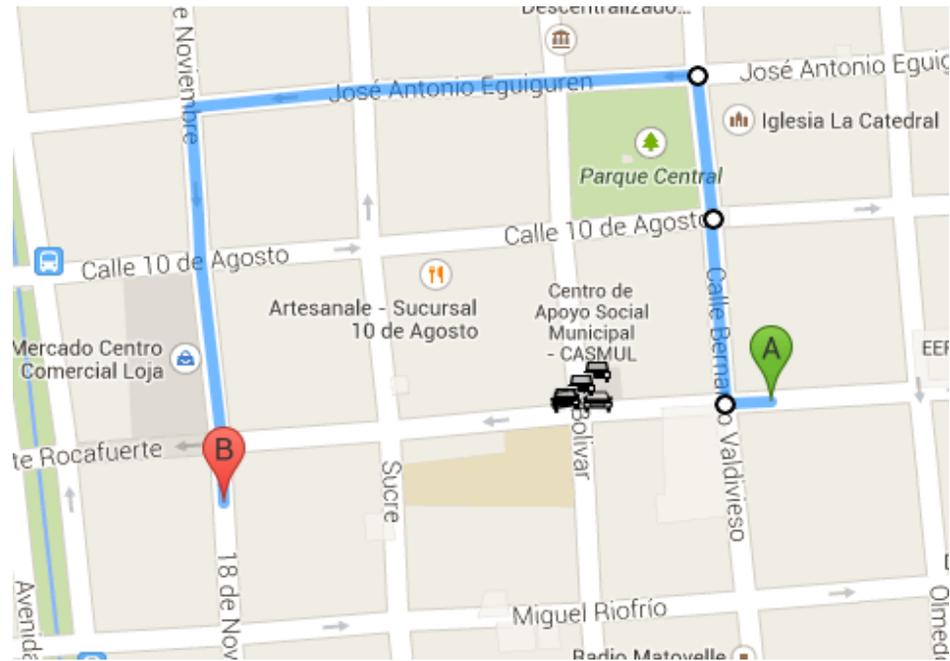


FIGURA 40. Prueba 4.

Prueba 5.

Accidente: Sucre y José Antonio Eguiguren, agente 035.

Lugar de origen A: Sucre y 10 de Agosto.

Lugar de destino B: Imbabura y Bolívar.

Si evita el Accidente.

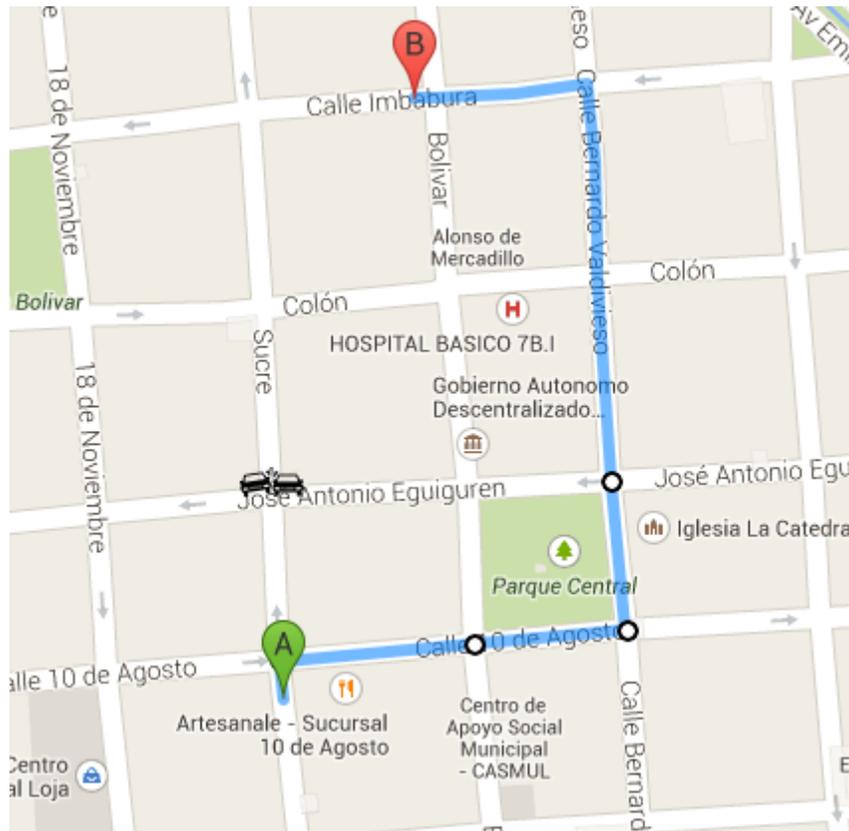


FIGURA 41. Prueba 5.

Prueba 6.

Accidente: Bolívar y Mercadillo, agente 081.

Lugar de origen A: Bolívar y Azuay.

Lugar de destino B: Catacocha y Sucre.

Si evita el Accidente.

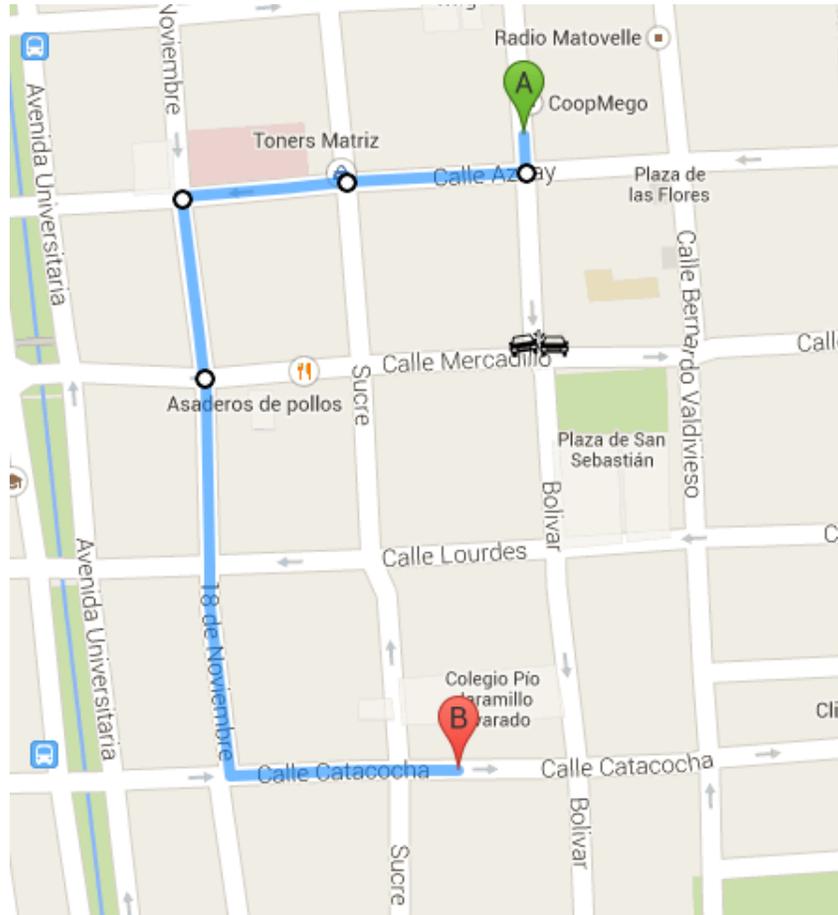


FIGURA 42. Prueba 6.

Prueba 7.

Accidente: Bolívar y Rocafuerte, agente 054.

Lugar de origen A: 10 de Agosto y Bolívar.

Lugar de destino B: Azuay y Bolívar.

Si evita el Accidente.

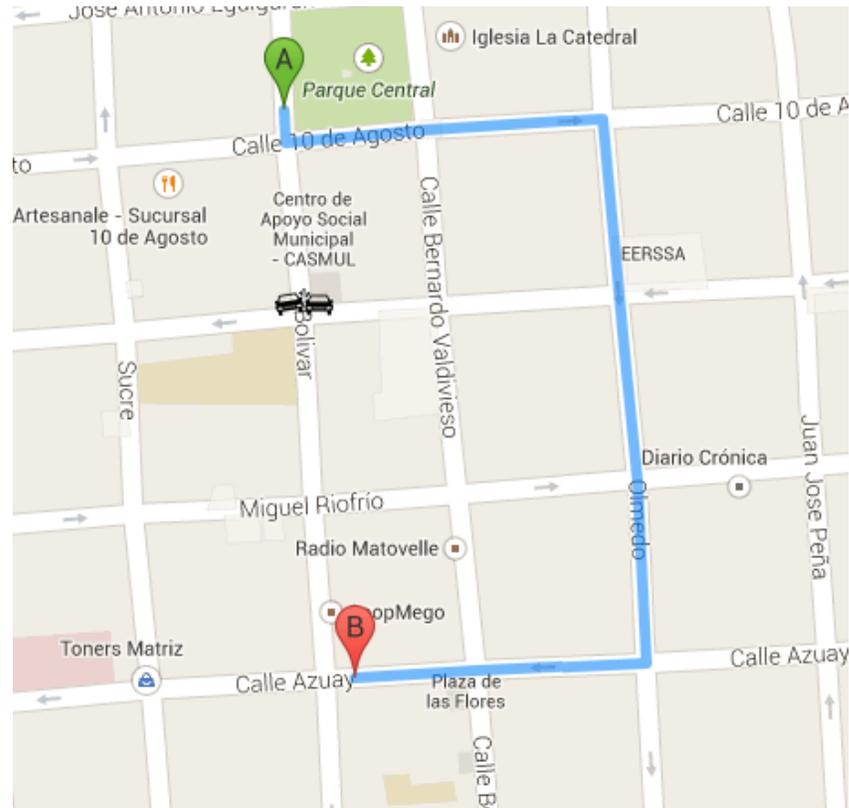


FIGURA 43. Prueba 7.

Prueba 8.

Accidente: 10 de Agosto y Sucre, agente 042.

Congestionamiento: Rocafuerte y Bolívar, agente 054.

Lugar de origen A: 18 de Noviembre y José Antonio Eguiguren.

Lugar de destino B: Rocafuerte y Juan José Peña.

Si evita el Accidente y Congestionamiento.

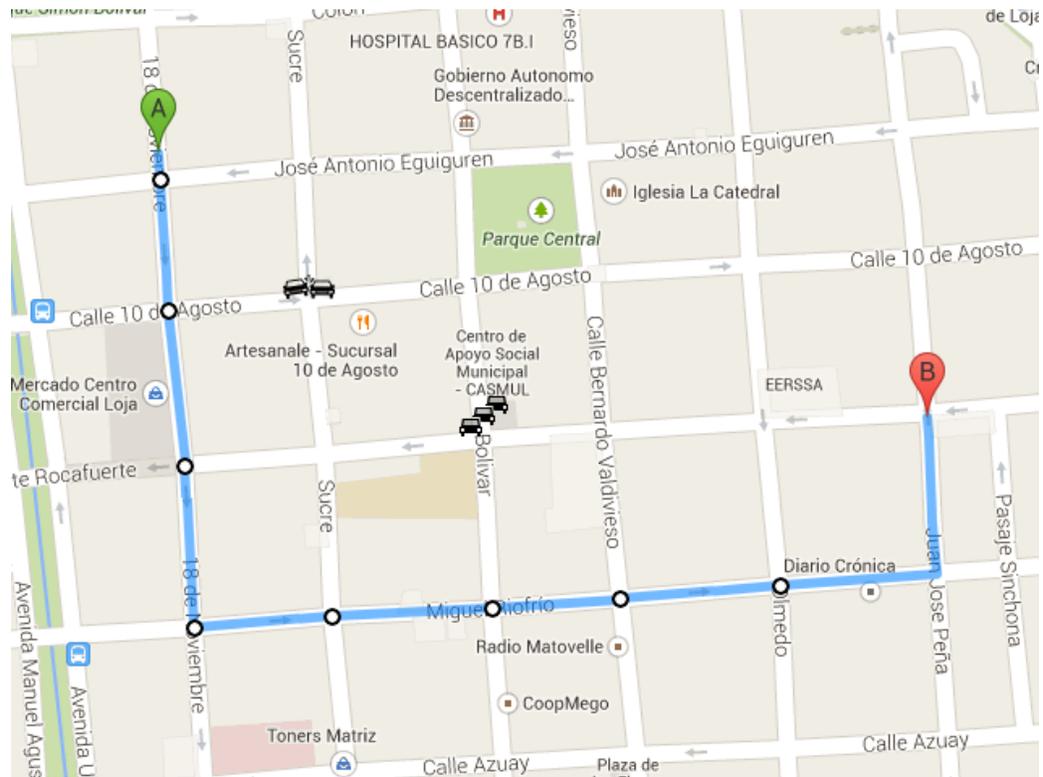


FIGURA 44. Prueba 8.

Prueba 9.

Accidente: 18 de Noviembre y Miguel Riofrío, agente 064:

Congestionamiento: Sucre y 10 de Agosto, agente 042:

Lugar de origen A: Mercadillo y Olmedo.

Lugar de destino B: José Antonio Eguiguren y Av. Universitaria.

Si evita el Accidente y Congestionamiento.

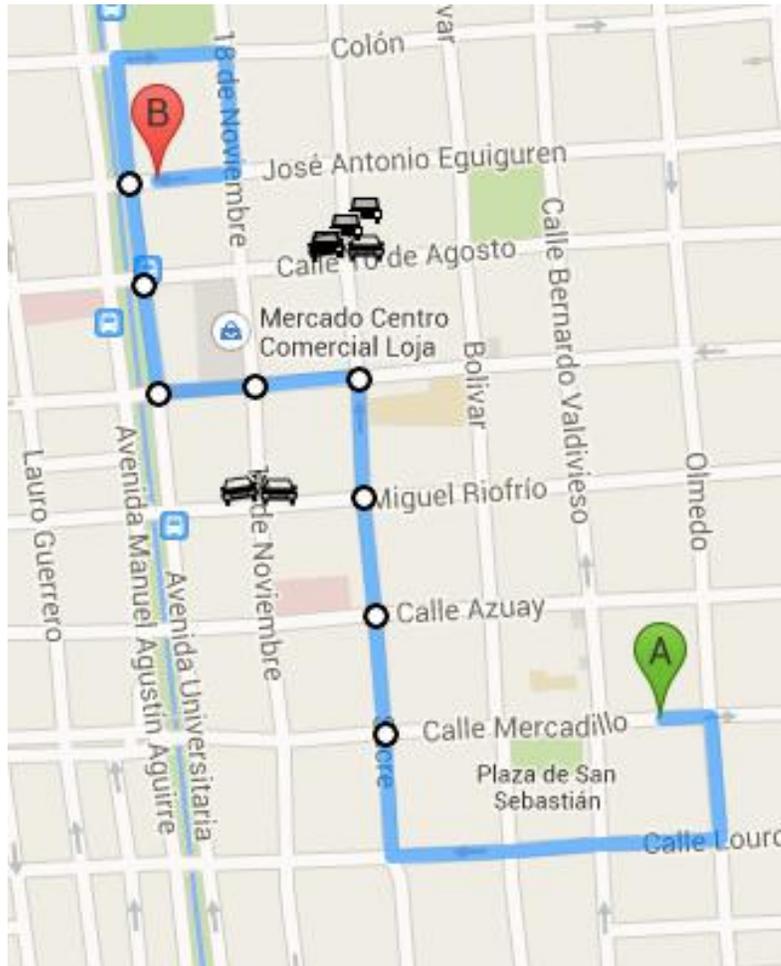


FIGURA 45. Prueba 9.

Prueba 10.

Accidente: 18 de Noviembre y Miguel Riofrío, agente 064.

Congestionamiento: Sucre y 10 de Agosto, agente 042.

Lugar de origen A: Bernardo Valdivieso y Azuay.

Lugar de destino B: José Antonio Eguiguren y Sucre.

Si evita el Accidente y Congestionamiento.

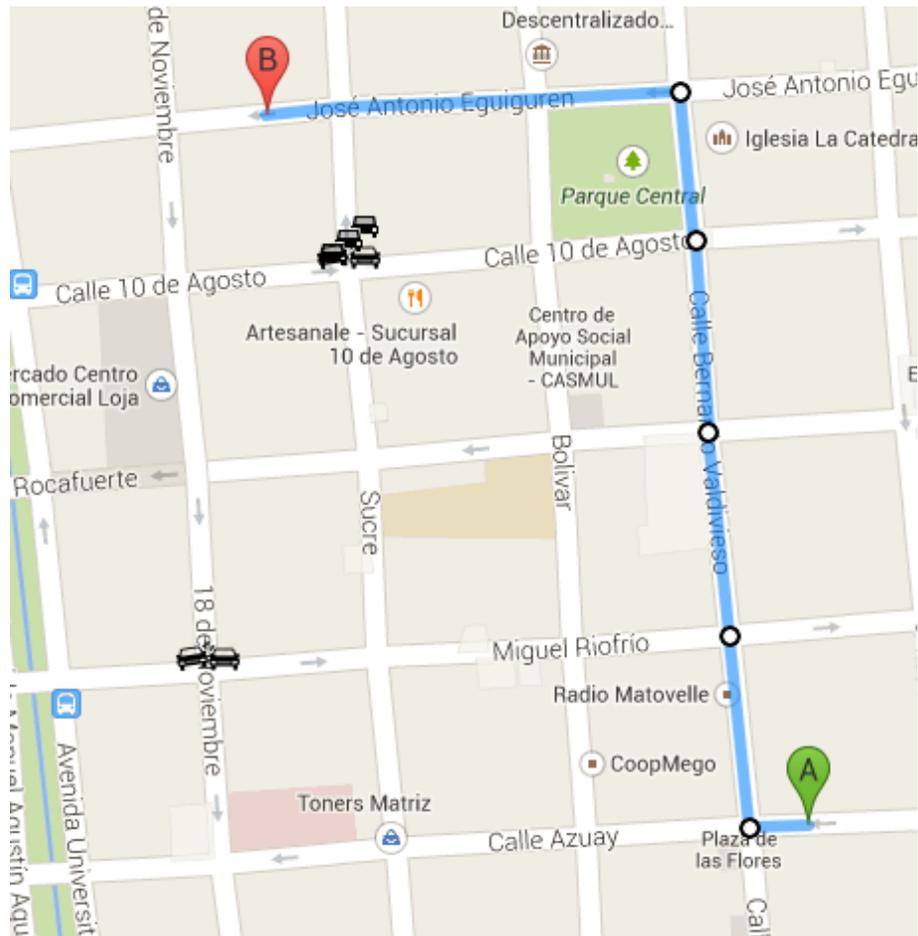


FIGURA 46. Prueba 10.

Prueba 11.

Accidente: Sucre y 10 de Agosto, agente 042.

Congestionamiento: Sucre y Miguel Riofrío, agente 065.

Lugar de origen A: Sucre y Azuay.

Lugar de destino B: Sucre y Colón.

Si evita el Accidente y Congestionamiento.

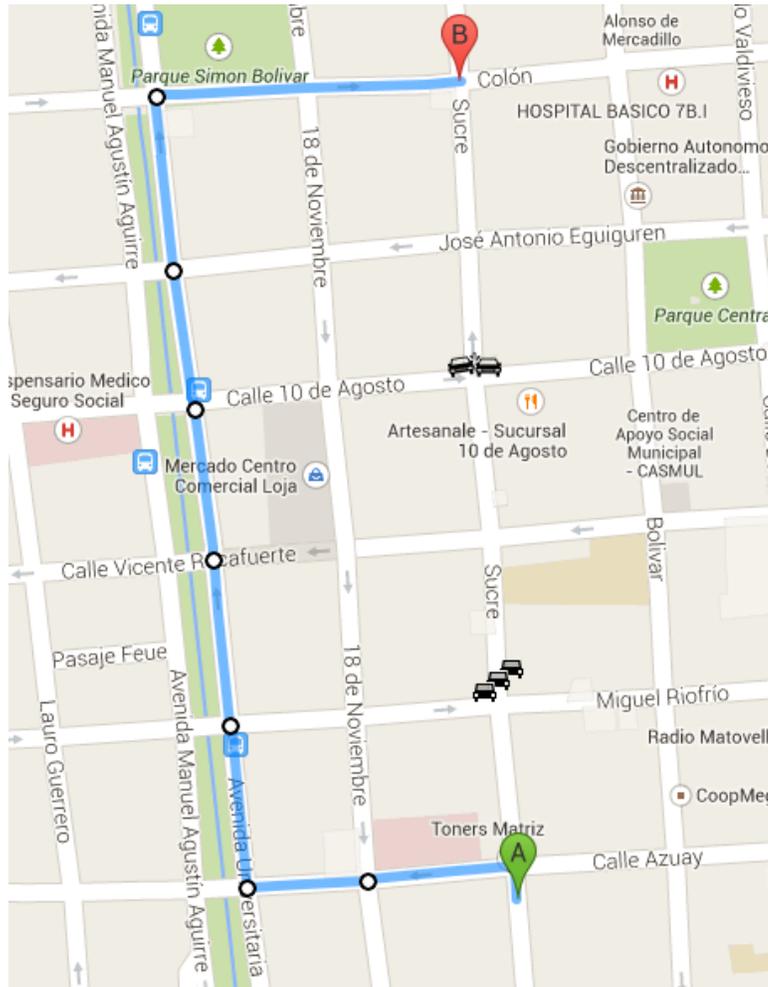


FIGURA 47. Prueba 11.

Prueba 12.

Accidente: Bernardo Valdivieso y 10 de Agosto, agente 044.

Accidente: Bernardo Valdivieso y Miguel Ríofrío, agente 067

Lugar de origen A: Azuay y Bernardo Valdivieso.

Lugar de destino B: José Antonio Eguiguren y Bolívar.

Si evita los 2 Accidentes.

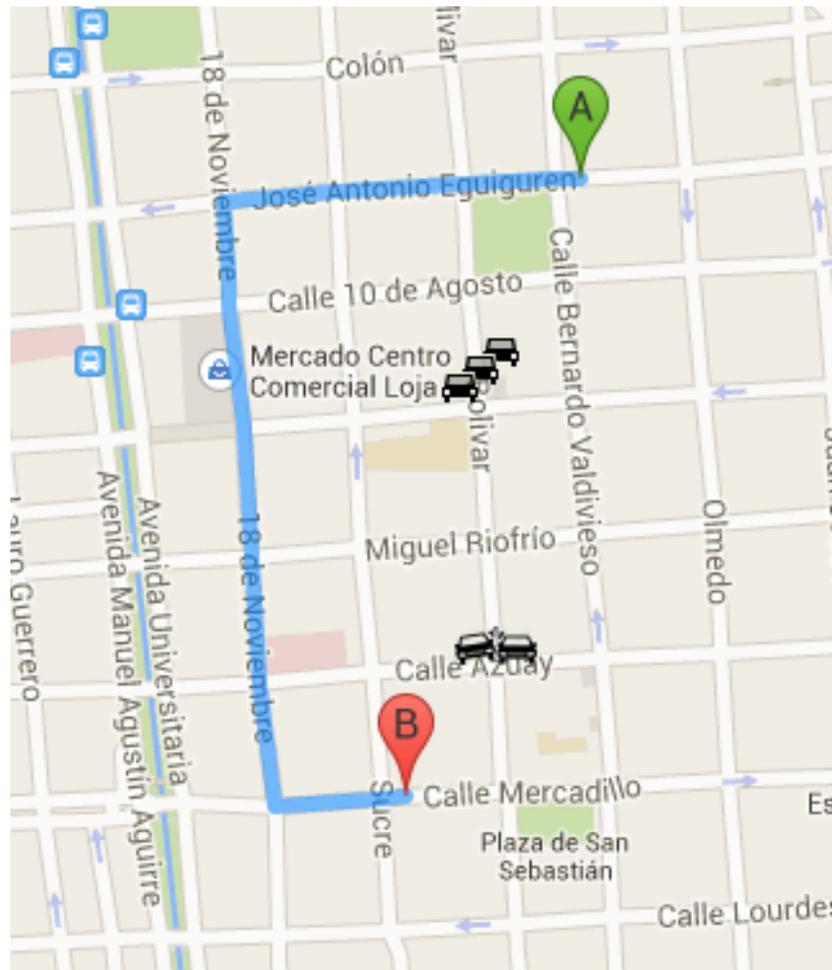


FIGURA 49. Prueba 13.

Prueba 14.

Accidente: Rocafuerte y Bernardo Valdivieso, agente 055.

Congestionamiento: Rocafuerte y Sucre, agente 053.

Lugar de origen A: Rocafuerte y Olmedo.

Lugar de destino B: Rocafuerte y AV. Universitaria.

Si evita el Accidente y Congestionamiento.

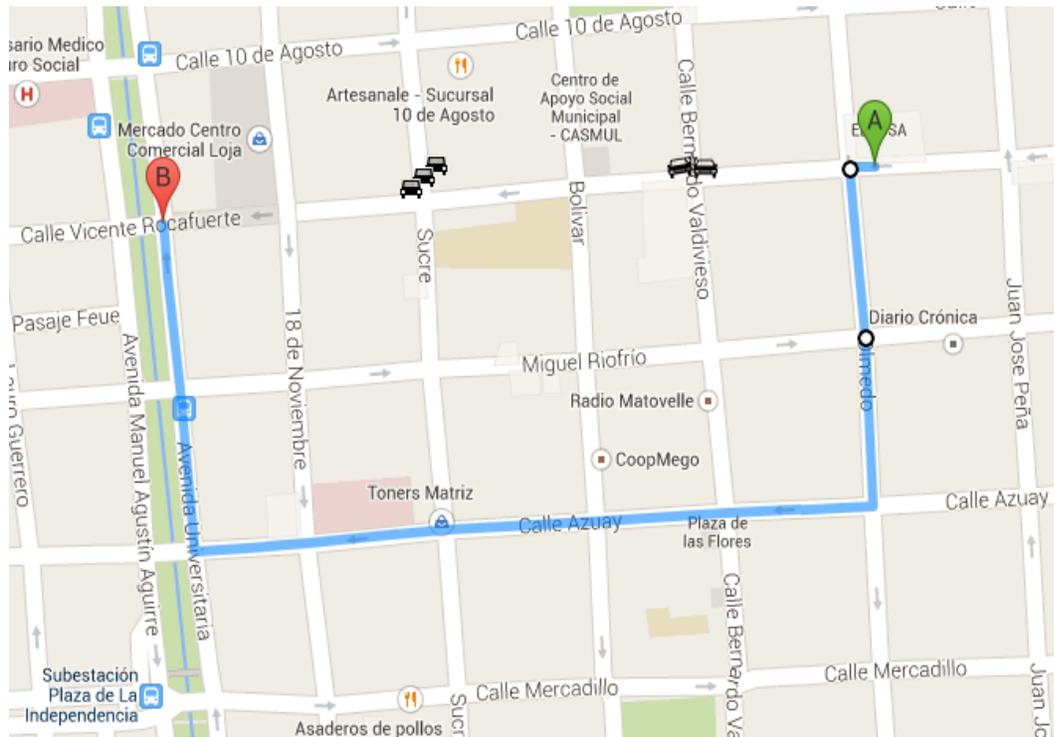


FIGURA 50. Prueba 14.

Prueba 15.

Accidente: 18 de Noviembre y 10 de Agosto, agente 041.

Congestionamiento: 18 de Noviembre y Rocafuerte, agente 129.

Lugar de origen A: José Antonio Eguiguren y 18 de Noviembre.

Lugar de destino B: Miguel Riofrío y 18 de Noviembre.

Si evita el Accidente y Congestionamiento.

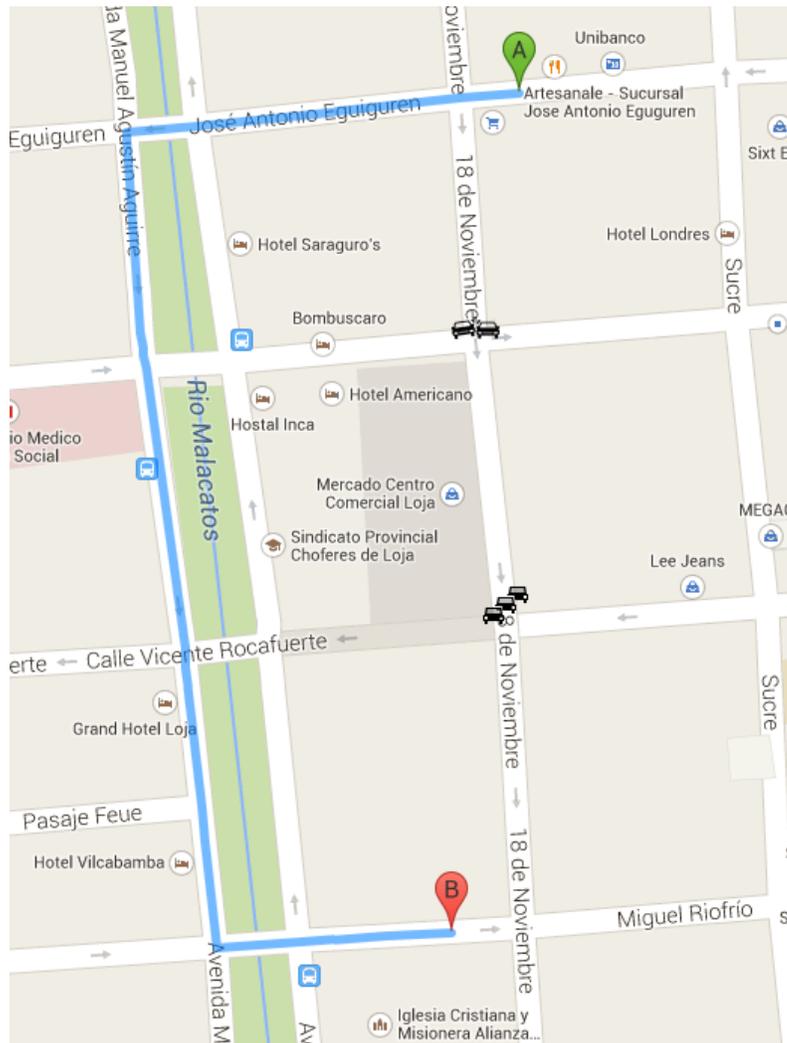


FIGURA 51. Prueba 15.

Prueba 16.

Congestionamiento: Miguel Riofrío y 18 de Noviembre, agente 064.

Congestionamiento: Miguel Riofrío y Bolívar, agente 066.

Lugar de origen A: Miguel Riofrío y Av. Universitaria.

Lugar de destino B: Azuay y Bolívar.

Si evita los 2 Congestionamientos.

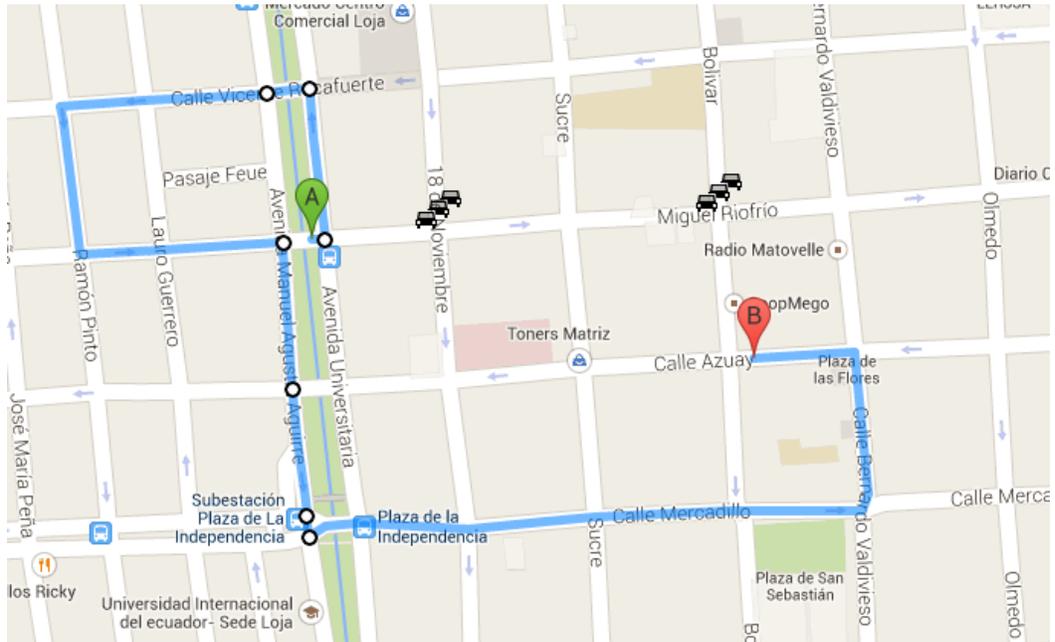


FIGURA 52. Prueba 16.

Prueba 17.

Accidente: 10 de Agosto y 18 de Noviembre, agente 041.

Congestionamiento: 18 de Noviembre y Rocafuerte, agente 129.

Accidente: 18 de Noviembre y Miguel Riofrío, agente 064

Lugar de origen A: José Antonio Eguiguren y Sucre.

Lugar de destino B: 18 de Noviembre y Mercadillo.

Si evita los 2 Accidentes y el Congestionamiento.

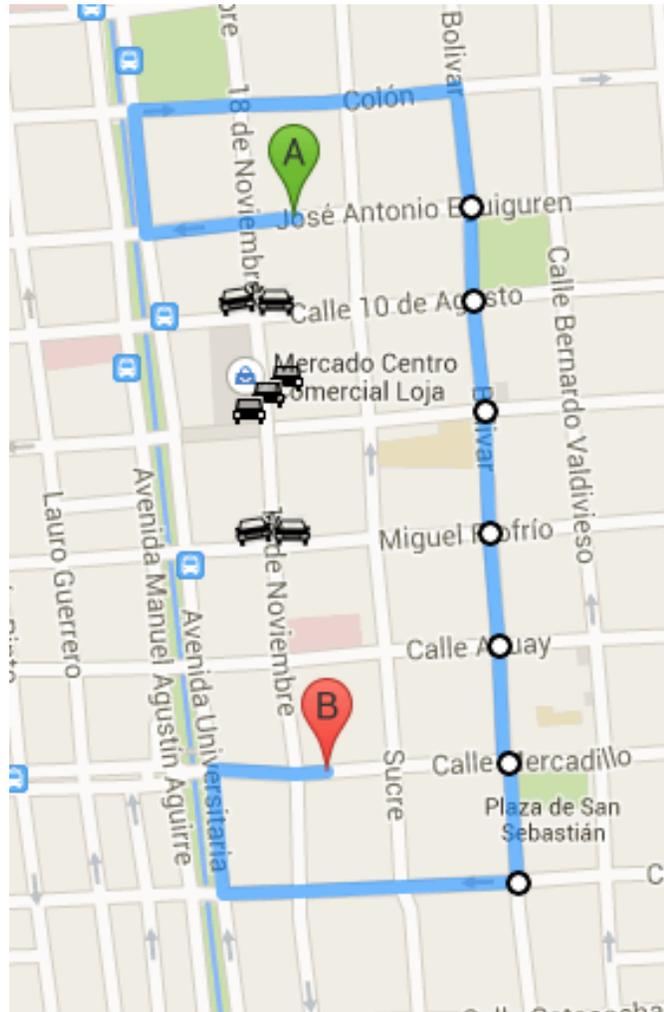


FIGURA 53. Prueba 17.

Prueba 18.

Accidente: José Antonio Eguiguren y Sucre, agente 035.

Accidente: Miguel Riofrío y Sucre, agente 065.

Congestionamiento: Av. Universitaria y Azuay, agente 063.

Lugar de origen A: Bolívar y Mercadillo.

Lugar de destino B: 18 de Noviembre y Rocafrío.

Si evita los 2 Accidentes y el Congestionamiento.

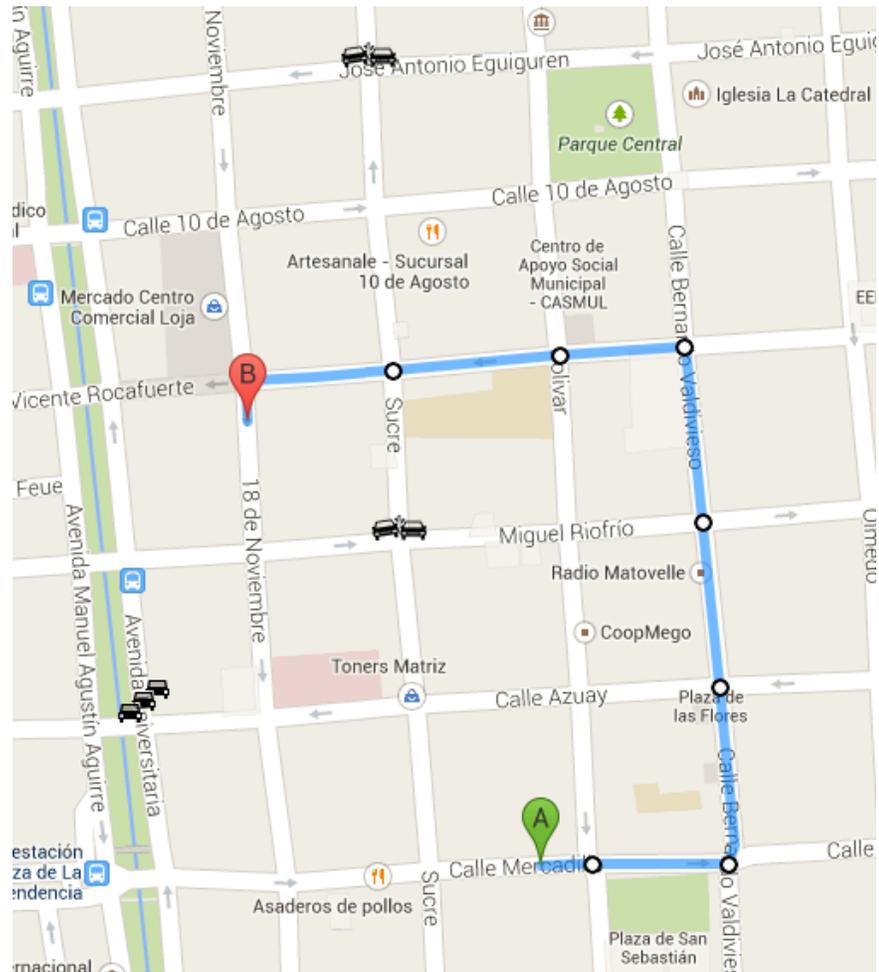


FIGURA 54. Prueba 18.

Prueba 19.

Accidente: Sucre y Miguel Riofrío, agente 065.

Congestionamiento: Bolívar y Rocafuerte, agente 054.

Congestionamiento: Sucre y 10 de Agosto, agente 042.

Lugar de origen A: Bernardo Valdivieso y Azuay.

Lugar de destino B: José Antonio Eguiguren y Bolívar.

Si evita los 2 Congestionamientos y el Accidente.

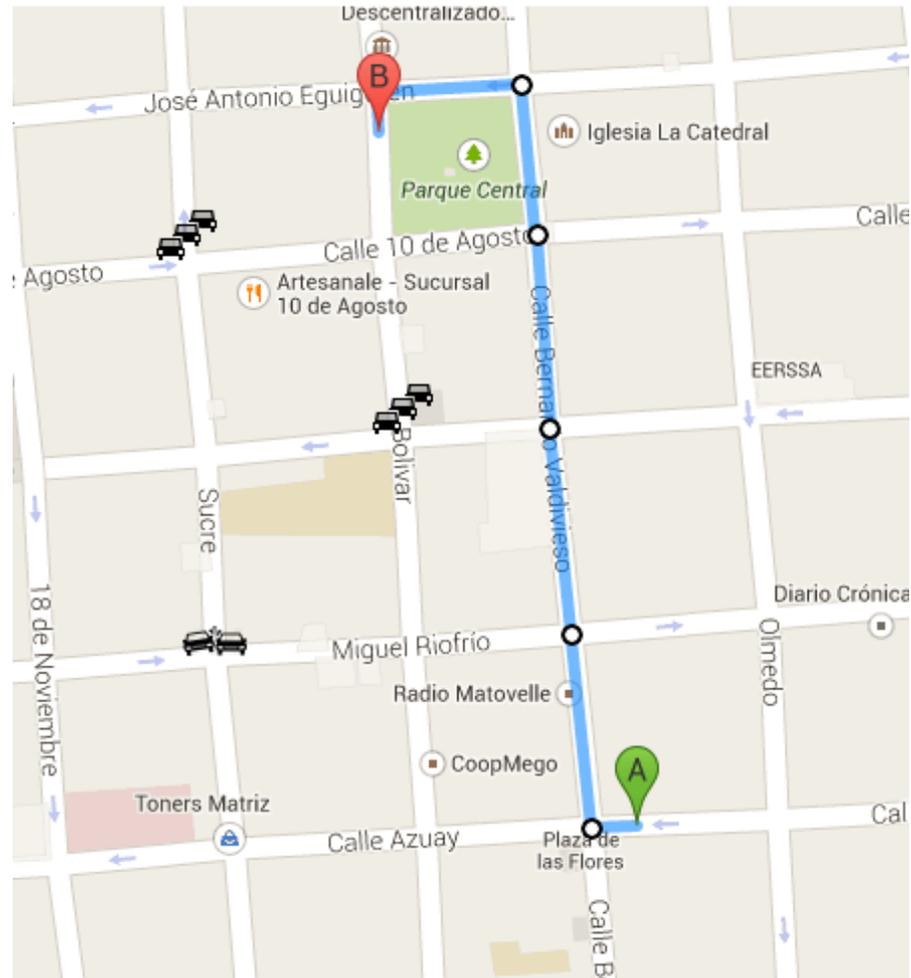


FIGURA 55. Prueba 19.

Prueba 20.

Accidente: Sucre y Miguel Riofrío, agente 065.

Congestionamiento: Bolívar y Rocafuerte, agente 054.

Congestionamiento: Sucre y 10 de Agosto, agente 042.

Lugar de origen A: Rocafuerte y Bernardo Valdivieso.

Lugar de destino B: 18 de Noviembre y Rocafuerte.

Si evita los 2 Congestionamientos y el Accidente.

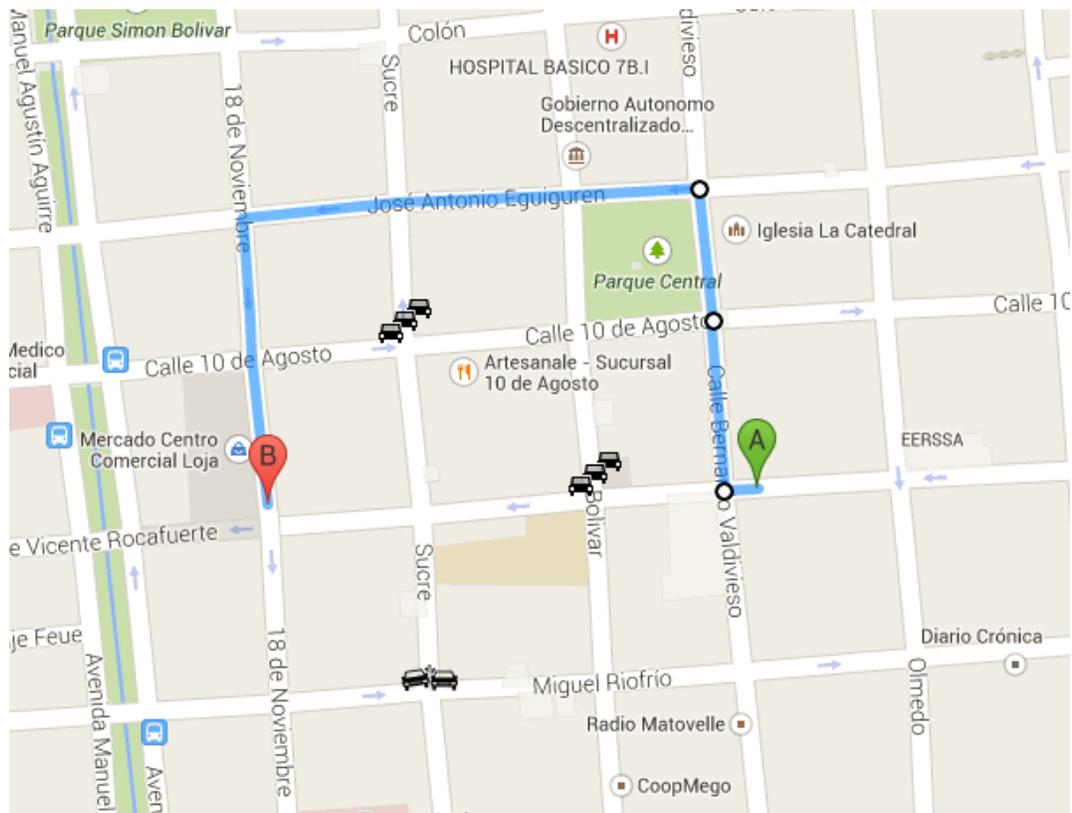


FIGURA 56. Prueba 20.

Prueba 21.

Accidente: Sucre y Miguel Riofrío, agente 065.

Congestionamiento: Bolívar y Rocafuerte, agente 054.

Congestionamiento: Sucre y 10 de Agosto, agente 042.

Lugar de origen A: José Antonio Eguiguren y Bernardo Valdivieso.

Lugar de destino B: 18 de Noviembre y Azuay.

Si evita los dos Congestionamientos y el Accidente.

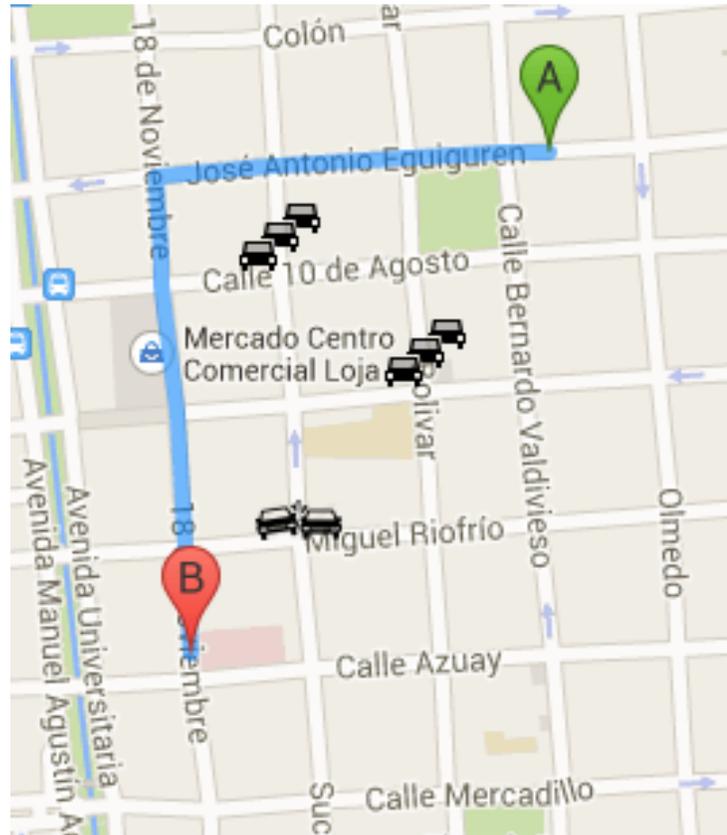


FIGURA 57. Prueba 21.

Prueba 22.

Accidente: 18 de Noviembre y Rocafuerte, agente 129.

Accidente: Bolívar y Rocafuerte, agente 054.

Accidente: Sucre y Miguel Riofrío, agente 065.

Accidente: Bernardo Valdivieso y Miguel Riofrío, agente 067.

Congestionamiento: Miguel Riofrío y Bolívar, agente 066.

Congestionamiento: Bolívar y Azuay, agente 087.

Lugar de origen A: Bolívar y 10 de Agosto.

Lugar de destino B: Lourdes y Olmedo.

Si evita los 2 Congestionamientos y 4 Accidentes.

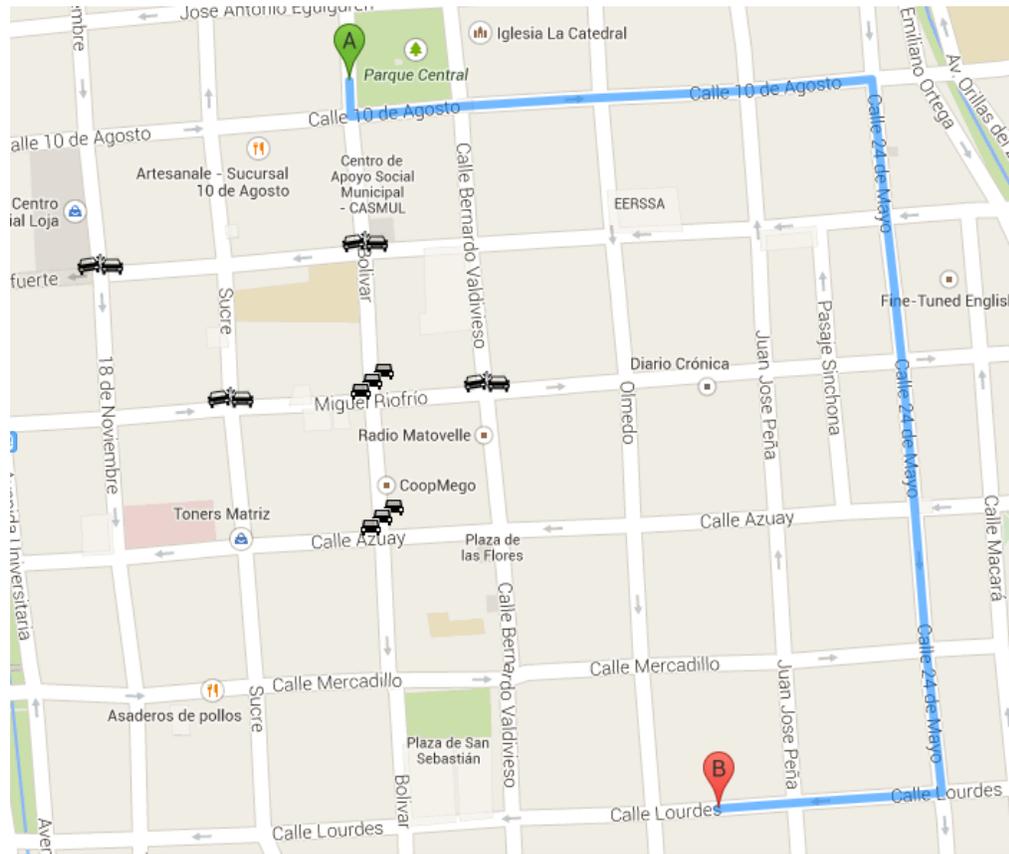


FIGURA 58. Prueba 22

Prueba 23.

Accidente: 18 de Noviembre y Rocafuerte, agente 129.

Accidente: Bolívar y Rocafuerte, agente 054.

Accidente: Sucre y Miguel Riofrío, agente 065.

Accidente: Bernardo Valdivieso y Miguel Riofrío, agente 067.

Congestionamiento: Miguel Riofrío y Bolívar, agente 066.

Congestionamiento: Bolívar y Azuay, agente 087.

Congestionamiento: Bernardo Valdivieso y Azuay, agente 069.

Lugar de origen A: Bolívar y 10 de Agosto.

Lugar de destino B: Bolívar y Lourdes.

Si evita los 3 Congestionamientos y 4 Accidentes.

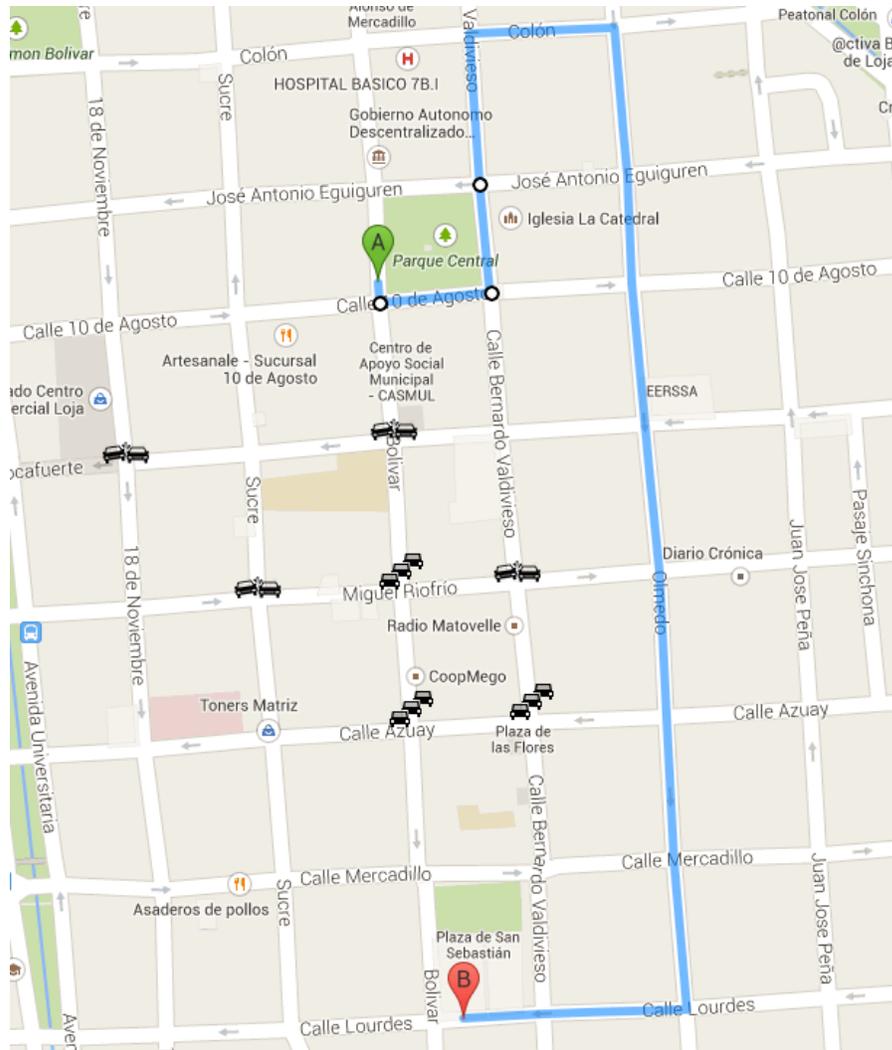


FIGURA 59. Prueba 23.

Prueba 24.

Accidente: Bolívar y Miguel Riofrío, agente 066.

Accidente: Bolívar y Azuay, agente 087.

Congestionamiento: José Antonio Eguiguren y Sucre, agente 035.

Congestionamiento: 18 de Noviembre y 10 de Agosto, agente 041.

Congestionamiento: Bolívar y 10 de Agosto, agente 043.

Congestionamiento: Sucre y Rocafuerte, agente 053.

Prueba 26.

Congestionamiento: José Félix de Valdivieso y Av. Universitaria, agente 019.

Congestionamiento: Colón y Av. Manuel Agustín Aguirre, agente 027.

Congestionamiento: 10 de Agosto y Av. Manuel Agustín Aguirre, agente 039.

Congestionamiento: Av. Universitaria y Rocafuerte, agente 050.

Congestionamiento: Av. Manuel Agustín Aguirre y Miguel Riofrío, agente 060.

Congestionamiento: Av. Manuel Agustín Aguirre y Azuay, agente 062.

Congestionamiento: 18 de Noviembre y Miguel Riofrío, agente 064.

Congestionamiento: Bolívar y Rocafuerte, agente 054.

Accidente: 18 de Noviembre y José Antonio Eguiguren, agente 034.

Accidente: Lauro Guerrero Rocafuerte, agente 059.

Lugar de origen A: Av. Manuel Agustín Aguirre y Av. Cuxibamba.

Lugar de destino B: Av. Manuel Agustín Aguirre y Mercadillo.

No evita los 8 Congestionamientos y 2 Accidentes, ruta demasiado extensa.

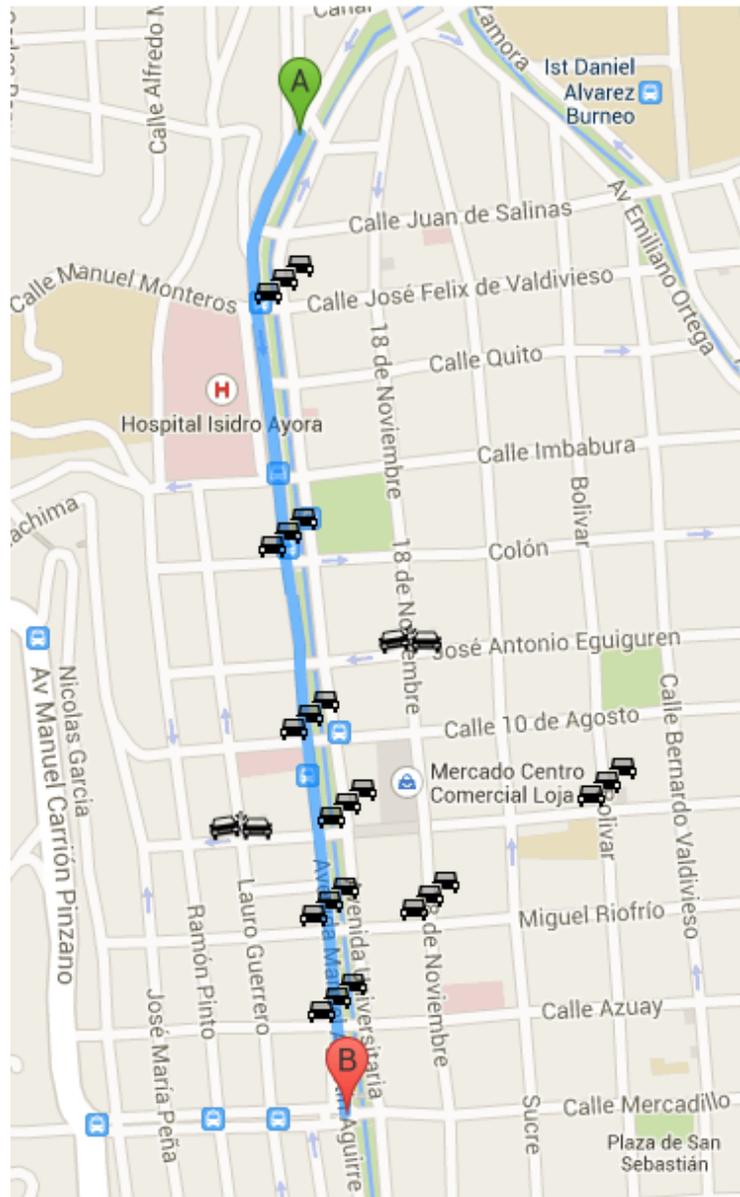


FIGURA 62. Prueba 26.

Prueba 27.

Congestionamiento: José Félix de Valdivieso y Av. Universitaria, agente 019.

Congestionamiento: Colón y Av. Manuel Agustín Aguirre, agente 027.

Congestionamiento: 10 de Agosto y Av. Manuel Agustín Aguirre, agente 039.

Congestionamiento: Av. Universitaria y Rocafuerte, agente 050.

Prueba 28.

Accidente: Bolívar y Rocafuerte, agente 054.

Lugar de origen A: Bolívar y Rocafuerte.

Lugar de destino B: Bolívar y Azuay.

No evita los el accidente punto de origen cerca al lugar del accidente.

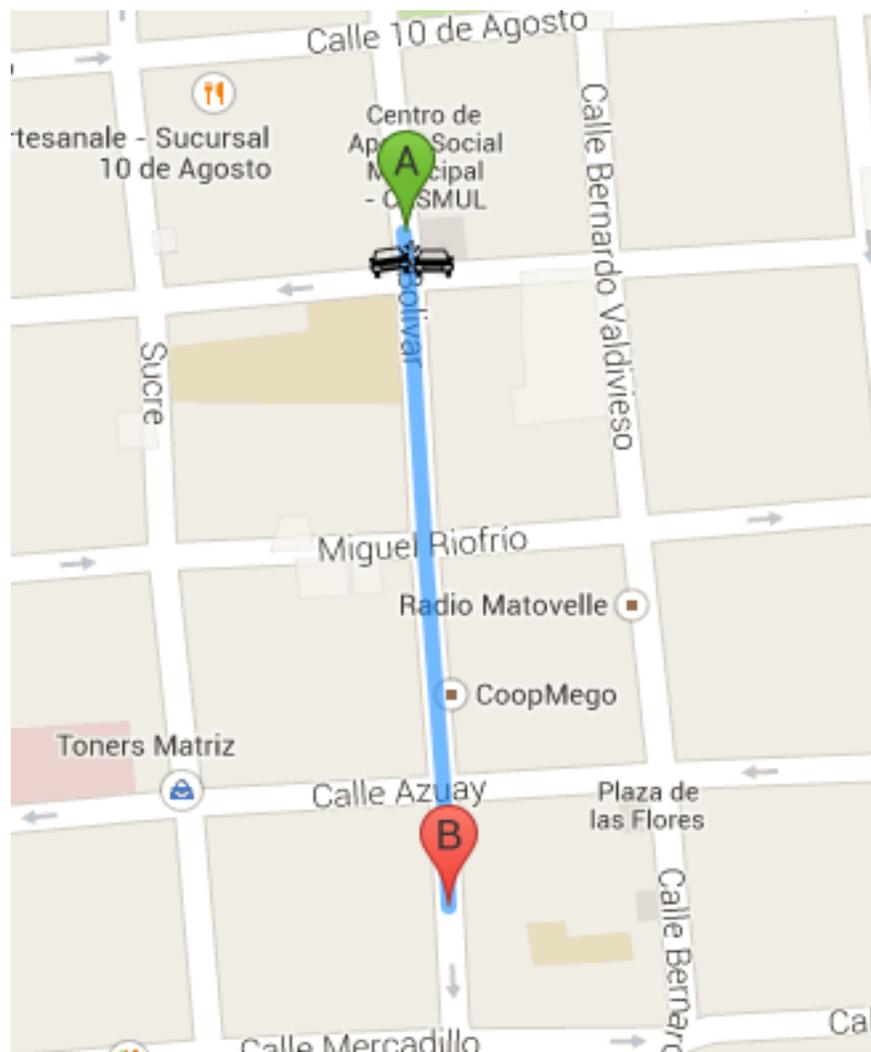


FIGURA 64. Prueba 28.

3.3. Realizar pruebas de factibilidad del sistema.

Las pruebas de factibilidad las realizamos de los valores obtenidos con las pruebas de funcionabilidad del sistema.

Se realizaron 28 pruebas del sistema las mismas fueron hechas en la parte donde existen la mayor cantidad de agentes, los eventos fueron producidos en diferentes intersecciones de la parte céntrica de la ciudad de Loja y se obtuvo los siguientes datos los mismo que se describen en la siguiente tabla.

TABLA VII. RESULTADOS PRUEBAS

Estado de prueba	Cantidad	Porcentaje
éxito	25	89%
fallida	3	11%
Total pruebas	28	100%

Se obtuvieron 25 pruebas que tuvieron éxito al momento de trazar la ruta y 3 pruebas que resultaron fallidas.

Las 25 pruebas exitosas fueron realizadas en la parte céntrica de la ciudad con distancias cortas y procurando realizarlas donde existen agentes.

Las 3 pruebas que resultaron fallidas se tomaron a 2 de ellas trazando los puntos de inicio y fin con distancias extensas (Prueba 26 y Prueba 27), la misma que al trazar la ruta la traza sin tomar en cuenta los eventos q se han producido en los agentes y por ende en los nodos q intervienen por los eventos producidos. La otra prueba fallida (Prueba 28) tuvo ese estado ya que al momento de indicar el origen de la ruta se lo hizo cerca al agente donde se produjo un evento y por ende traza la ruta sin tomar en cuenta ningún nodo ya que intervienen en el evento producido.

De acuerdo a los resultados obtenidos podemos decir que el programa es factible para utilizarlo en la ciudad de Loja ya que de un total de 28 pruebas realizadas equivalente al

100%, el 89% resultaron exitosas y un 11% resultaron fallidas como se muestra en la siguiente figura.

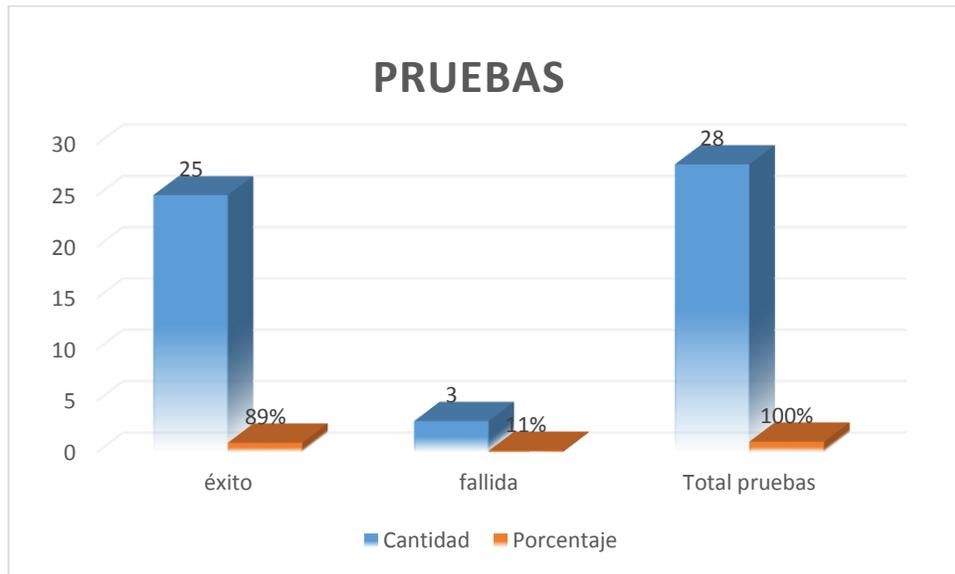


FIGURA 65. Resultados.

3.4. Presupuesto Implementación

A continuación se realiza el presupuesto con dos alternativas para el proyecto denominado “Desarrollo y simulación de un sistema multiagente para la comunicación de semáforos para encontrar la ruta óptima mediante grafos”.

En la primera alternativa se realiza el presupuesto con los semáforos que existen actualmente, con tecnología LED y que funcionan basado en el uso de controlador. Para la interconexión y comunicación de los semáforos se presupuesta el tendido de fibra óptica, para realizar la detección de accidentes y aglomeraciones de una manera eficaz se propone utilizar cámaras IP de alta definición.

En la segunda alternativa de igual manera se realiza el presupuesto con los semáforos led que está en función actualmente y que funcionan basado en el uso de controlador. Para la interconexión y comunicación de los semáforos a través de la red se propone el uso de AP's

(Puntos de acceso) que permitirá la comunicación de semáforos inalámbricamente evitando el tendido de fibra óptica, para realizar la detección de accidentes y aglomeraciones de una manera eficaz se propone utilizar cámaras IP de alta definición.

Para calcular la cantidad de fibra óptica que se utilizará, se tomó en cuenta los datos obtenidos en la página web del Gobierno autónomo descentralizado de la ciudad de Loja a través del proyecto Plan de ordenamiento y desarrollo sostenible del casco urbano central de la ciudad de Loja conocido como “Regeneración Urbana” que se encuentra en fases de desarrollo.

3.4.1. Topología física y lógica propuesta.

En caso de la implementación de fibra óptica se plantea utilizar la topología en Anillo tanto a nivel físico y lógico ya que permite contar con redundancias ante la presencia de daños en la red y así poder mantener el servicio activo, está topología provee un acceso equitativo para todos los dispositivos activos de la red.

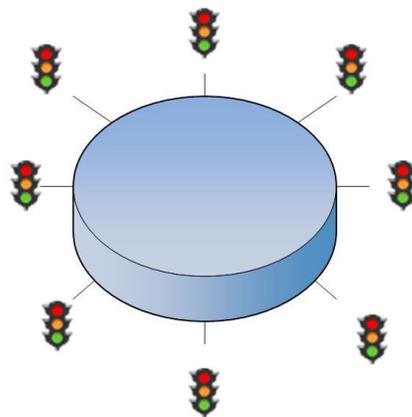


FIGURA 66. Topología en anillo.

3.4.2. Características de la alternativa con fibra óptica.

A continuación se visualiza el recorrido de fibra óptica alrededor de la ciudad de Loja, el total es de 23.409 Kilómetros, el tendido de fibra óptica es subterránea y para proteger la fibra óptica de roedores, agua, humedad se ha tomado en cuenta el uso de tubería de polietileno.



FIGURA 67. Tendido de Fibra Óptica.

El rollo de fibra o bovina de fibra óptica viene de 4 Kilómetros, para asegurar la continuidad de los hilos de fibra es necesario utilizar los equipos denominados “Manga de empalme” ya que icho equipo es el encargado de alojar y proteger las fusiones de los hilos de fibra óptica contra las condiciones de intemperie.

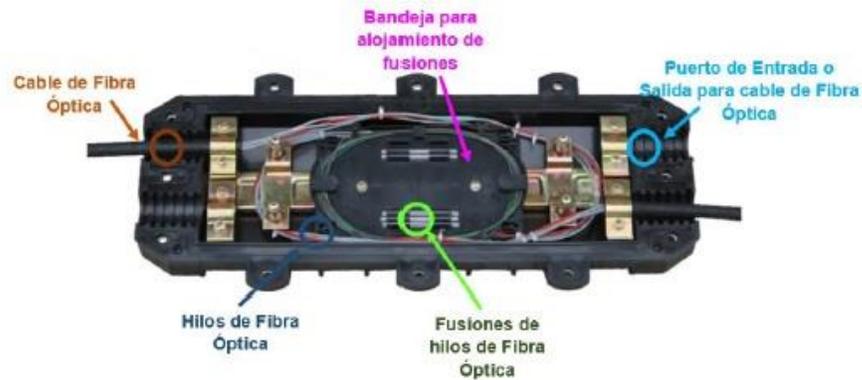


FIGURA 68. Manga de Empalme.

A continuación se muestra un ejemplo gráfico de la instalación de la manga de empalme para unir los hilos de fibra óptica.



FIGURA 69. Gráfico de instalación de Manga de Empalme.

Para la conexión de los equipos entre el cable de fibra óptica proveniente de la parte externa y los equipos de activos. Para los semáforos se instalaría en la intersección el ODF sería de plástico ya que como se encuentra a la intemperie será resistente a lluvias, etc.

El ODF posee uno o varios puertos de ingreso de cables y una área de patcheo en el cual se conecta la terminación del cable de fibra óptica por el un extremo, y el patch cord hacia el equipo activo por el otro extremo.



FIGURA 70. ODF plástico.

También se utilizara los patch cord cuya función es conectar el ODF con el equipo activo, se utilizara el pigtail que es un hilo de fibra óptica cuya función es fusionarse con un hilo de cable de fibra óptica de un extremo y conectarse a un adaptador del ODF, un pigtail cuenta con un conector en uno de sus extremos.

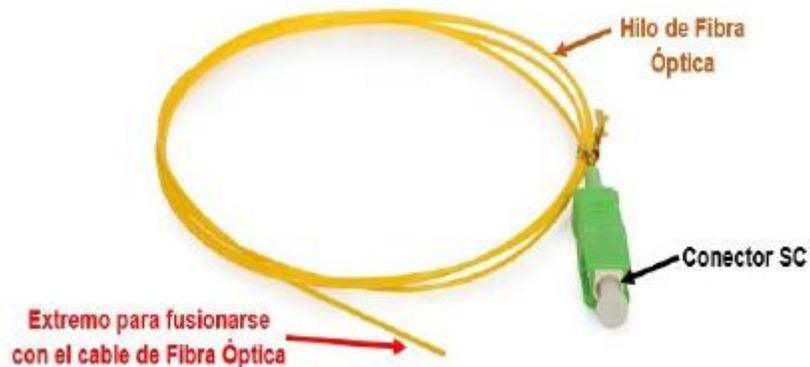


FIGURA 71. Pigtail de Fibra óptica.

Las pruebas de atenuación de fibra óptica permiten calcular la pérdida que puede tener un enlace de fibra óptica, así mismo permite establecer que la fibra funciona con los estándares establecidos.

La pérdida en las mangas de empalme está establecido máximo sobre el 0.5 dB/KM y por debajo del 0.1 db/KM.

TABLA VIII. PRESUPUESTO ALTERNATIVA 1

Concepto.	Unidad.	Cantidad.	Costo Unitario	Costo Total
EQUIPOS				
Cámaras IP HD CAMARA TRENDNET OUTDOOR POE DAY/NIGHT TV- IP302PI	Cámaras	117	244.16	28.566,72
Acondicionamiento Electrónico	(Cajas)	117	150	17.550,00
Fibra óptica monomodo para la interconexión de semáforos.	Kilómetros	23.409	1.300	30.431,70
Manga de empalme domo	Unidad	6	95,00	570,00
ODF plástico	ODF	117		

Patch cord de fibra SC-LC, monomodo 3 metros.	Patch cord	234	25,00	5.850,00
Pigtail de Fibra.	Pigtail	234	10,00	2.340,00
Switch de capa 2, 24 puertos.	Unidad	5	700,00	3.500,00
Router switch Cisco.	Unidad	2	560,00	1.120,00
SERVICIOS Y MANO DE OBRA				
Armado y montaje de ODF, manga de empalme	Personal	1	2.800	2.800,00
Pruebas de atenuación de fibra óptica.	Unidad	2	90,00	180,00
Tubería de Polietileno	Metros	14.128,80	4,27	60.329,98
Tendido de Fibra óptica	Metros	24.000	0,90	21.600,00
Armado de rack de fibra óptica y accesorios.	Unidad	1	220,00	220,00
Total				169.208,40

3.4.3. Características de la alternativa 2 con puntos de acceso.

Los puntos de acceso inalámbrico permiten distribuir una señal inalámbrica, permitiendo a dispositivos móviles o equipos que disponen de tarjetas de red inalámbricas su interconexión.

Es económico y permite a una ciudad, empresa distribuir si servicio ya sa de internet de una manera estética, sin gastar en cables, etc.

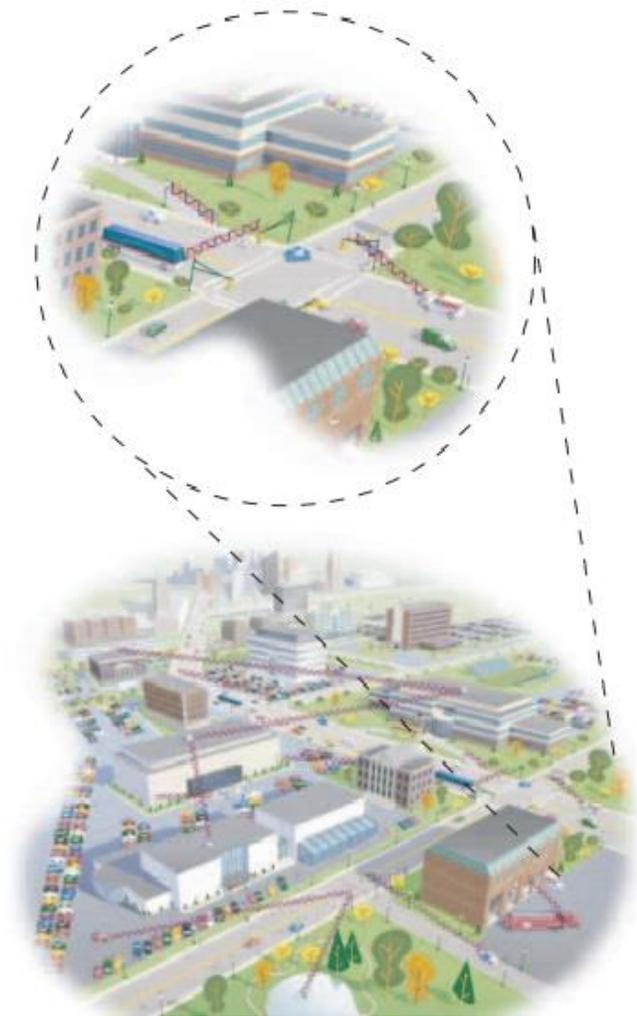


FIGURA 72. Ejemplo de Internet inalámbrico en una ciudad.

Para el desarrollar el presupuesto se ha cotizado Access Point que trasmitan la señal inalámbrica en 2.4 Ghz y que sirvan como conexiones Punto a Punto.

TABLA IX. PRESUPUESTO ALTERNATIVA 2.

Concepto.	Unidad.	Cantidad.	Costo Unitario	Costo Total
Equipos				
Cámaras IP HD CAMARA TRENDNET OUTDOOR POE DAY/NIGHT TV-IP302PI	Cámaras	117	\$ 244.16	28.566,72
Acondicionamiento Electrónico	(Cajas)	28	\$ 150	\$ 4.200
Access Point (Puntos de acceso.) Ubiquiti Networks AMO- 5G13 5Ghz 13Dbi Airmax Omni Antenna With a Rocket M BaseStation for 360°	Unidad	117	\$ 315.00	36.855
SERVICIOS				
Personal contratado	Personal	1	2.800	2.800
Total				74.421,72



FIGURA 73. Vista de semáforos funcionando con Access Point.

3.4.4. Ventajas y Desventajas de las alternativas propuestas

Alternativa 1.

Ventajas

- No existe pérdida de señal.
- Los semáforos se pueden comunicar de manera exitosa y sin retrasos.
- Es independiente de la situación geográfica de la ciudad, ya que el tendido de fibra óptica puede ser aéreo o de manera subterránea.
- El control de los semáforos es centralizado

Desventajas

- Mayor tiempo de implementación.
- Mantenimiento costoso de los equipos de interconexión, fibra óptica, ruteadores, etc.

Alternativa 2.

Ventajas.

- Menor costo de mantenimiento
- Instalación en un tiempo menor
- El control de los semáforos es centralizado.

Desventajas:

- Pérdida de señal
- Frecuencias saturadas
- Edificios que interfieren en la señal inalámbrica
- Pérdida de comunicación.

g. Discusión

1. Desarrollo De La Propuesta Alternativa

El presente proyecto de tesis denominado “**DESARROLLO Y SIMULACIÓN DE UN SISTEMA MULTIAGENTE PARA LA COMUNICACIÓN DE SEMÁFOROS PARA ENCONTRAR LA RUTA ÓPTIMA MEDIANTE GRAFOS**”, dio como resultado la construcción de la aplicación, desarrollado bajo el lenguaje de programación Java, y utilizando el Framework JADE que proporciona el soporte adecuado para la creación de Agentes, se utilizó el gestor de Base de datos MYSQL donde se almacenaran los semáforos y el mapa de la ciudad de Loja, a través de google maps . Se tomó en cuenta los semáforos que existen en la ciudad de Loja, con sus características y ubicación datos que fueron obtenidos por parte del municipio de Loja.

El objetivo general así como los objetivos específicos fueron alcanzados en su totalidad, gracias a los métodos y técnicas orientados al desarrollo y a la metodología GAIA que es una metodología para el diseño basada en agentes. A continuación evaluaremos los objetivos q fueron planteados en el proyecto de tesis

Objetivo Especifico 1: Desarrollar los agentes inteligentes para la comunicación de semáforos en tiempo real.

Dentro de este objetivo se realizó el estudio de los agentes inteligentes deliberativos, su funcionamiento, su interacción ante el entorno que lo rodea y su uso de técnicas de inteligencia artificial para conseguir mejor prestaciones.

Los algoritmos deliberativos ayudan a resolver tareas que requieren un modelo simbólico, donde las decisiones se basan en razonamientos lógicos, basada en la concordancia de patrones y la manipulación simbólica, utilizando la arquitectura deliberativa los semáforos tomarán las mejor decisión de manera autónoma y lógica cuando exista un accidente o aglomeración basada en la concordancia de patrones.

Se seleccionó las herramientas de software necesarias para facilitar el desarrollo de agentes, así como la metodología adecuada que permitió definir los roles y protocolos que realizará cada agente. Se utilizó el mapa de la ciudad de Loja a través de google maps como referencia con sus respectivos semáforos para poder simular accidentes y aglomeraciones y el usuario poder trazar una ruta para poder evitar dichos eventos..

Todo lo especificado anteriormente se detalla en la sección de resultados, apartado ¿? PRIMERA FASE: Desarrollar los agentes inteligentes para la comunicación de semáforos en tiempo real.

Objetivo Especifico 2: Desarrollar un algoritmo que permitan establecer la ruta más óptima en base a la ubicación de semáforos.

Para el desarrollo del algoritmo que le permita al usuario trazar una ruta evitando accidentes y aglomeraciones se analizó el funcionamiento de los algoritmos de Dijkstra ya que este permite trazar rutas de un origen a un destino por el camino más corto.

Analizado el algoritmo de dijkstra se procedió a diseñar y desarrollar el algoritmo basado en los semáforos como nodos y la distancia entre cada nodo o semáforo será el peso que el algoritmo tomara en cuenta, dicho algoritmo tendrá la capacidad de no tomar en cuenta los semáforos que está en un estado de accidente o aglomeración con el objetivo de evitar los mismos y así poder trazar una ruta dado un origen y un destino los más óptimo posible

Para analizar el proceso de diseño y creación del algoritmo, revisar la sección resultados, apartado ¿? SEGUNDA FASE: Desarrollar un algoritmo que permita establecer la ruta más óptima en base la ubicación de semáforos.

Objetivo Especifico 3: Integrar y simular agentes inteligentes y algoritmos de ruta óptimas para evitar Congestionamiento y/o accidentes.

Dentro de este objetivo unificamos en una aplicación web los agentes inteligentes y el algoritmo que permite establecer la ruta óptima, ayudándonos de herramientas que nos permitieron diseñar una interfaz amigable para el usuario el mismo q se va a valer de la

aplicación para simular la creación de agentes y a su vez trazar la ruta optima evitando los eventos producidos por el mismo.

Para analizar el proceso de diseño de la interfaz, revisar la sección resultados, apartado ¿? TERCERA FASE: Integrar y simular agentes inteligentes y algoritmo de rutas óptimas para evitar aglomeraciones.

Para el desarrollo de la aplicación **“DESARROLLO Y SIMULACIÓN DE UN SISTEMA MULTIAGENTE PARA LA COMUNICACIÓN DE SEMÁFOROS PARA ENCONTRAR LA RUTA ÓPTIMA MEDIANTE GRAFOS”**, se utilizó el Framework JADE q es desarrollado bajo el lenguaje de programación JAVA por ser multiplataforma hicieron y tener una sintaxis, clara y sencilla hizo que la construcción sea factible a través de su entorno NetBeans, además de utilizar google maps para la visualización del mapa de la ciudad de Loja.

2. Valoración técnica económica ambiental

2.1. Recursos

Para el **“DESARROLLO Y SIMULACIÓN DE UN SISTEMA MULTIAGENTE PARA LA COMUNICACIÓN DE SEMÁFOROS PARA ENCONTRAR LA RUTA ÓPTIMA MEDIANTE GRAFOS”**, se contó con los recursos humanos, técnicos, económicos y tecnológicos como hardware y software, lo cual permitió culminar con éxito el presente proyecto de tesis.

En cuanto al aspecto económico, los recursos materiales fueron adquiridos por los tesisistas, en su totalidad.

Por lo antes presentado, la ejecución del proyecto fue totalmente factible y dando cumplimiento a todos los objetivos planteados. Los recursos utilizados se detallan a continuación:

- **Recursos Humanos**

Para el desarrollo del proyecto se ha necesitado la ayuda profesional del director de tesis, quien ayudó en la dirección del proyecto siguiendo las guías institucionales. Debido al carácter investigativo y académico del proyecto, este se llevó a cabo por dos egresados de la Carrera de Ingeniería en Sistemas, que hicieron las veces de analistas y programadores.

La tabla 8 refleja el número de horas que han empleado cada uno de estos perfiles en el desarrollo del proyecto. Este número de horas de trabajo multiplicado por el coste por hora de cada uno, da como resultado el coste total del recurso humano.

TABLA X. RECURSOS HUMANOS.

TALENTOS HUMANOS			
EQUIPO TRABAJO	TIEMPO (HORAS)	PRECIO/HORA (\$)	VALOR TOTAL
Robert Iván Loarte Caraguay	640	8,00	\$5120,00
Bolívar Rolando Quizhpe Vásquez	640	8,00	\$5120,00
Asesor 1	240	0,0	\$0,00
SUBTOTAL			\$10240,00

- **Recursos Materiales**

Para el desarrollo del proyecto ha sido necesario el uso de material de oficina empleado para la toma de notas auxiliares, almacenamiento de la información y documentación final.

Como resultado se presenta la siguiente tabla que indica los costes de material desglosados:

TABLA IXI. RECURSOS MATERIALES.

RECURSOS MATERIALES			
RUBRO	CANT.	UNIDAD	VALOR TOTAL
Hojas A4	300	0.03	\$9.00
Anillado	3	2.50	\$7.50
Perfil	6	0.75	\$4.50
CD's	3	0.50	\$1.50
Dvd's	2	0.75	\$1.50
SUBTOTAL			\$ 24.00

- **Recursos Técnicos/Tecnológicos**

Estos recursos fueron divididos en tres secciones: Recursos de Hardware, software y Comunicaciones, a continuación los detallamos:

- **Recursos de Hardware**

El uso de equipos es parte de los materiales usados en el desarrollo del presente proyecto, los cuales son indicados en la siguiente tabla:

TABLA XII. RECURSOS DE HADWARE.

RECURSOS FÍSICOS					
RUBROS	CANT.	VALOR	TIEMPO UTILIZADO (MESES)	PRECIO /MES	VALOR TOTAL
Portátil Dell i5	1	\$1100.00	12	\$26.66	\$320.00
Portátil Samsung i5	1	\$1100.00	12	\$26.66	\$320.00
Impresora	1	\$60.00	1	\$12.00	\$12.00
SUBTOTAL					\$ 652.00

- **Recursos de Software**

El software que usamos para el desarrollo de la aplicación es libre, por tal razón no se desembolsó ninguna cantidad de dinero. En la siguiente tabla se detalla el software utilizado:

TABLA XIII. RECURSOS DE SOFTWARE.

RECURSOS SOFTWARE			
RUBRO	CANT.	UNIDAD	VALOR TOTAL
Sistemas Operativo Ubuntu 12.04	1	0.00	\$0.00
Herramienta de Programación de java (NetBeans 8.01)	1	0.00	\$0.00
Herramienta de Programación Base de Datos (MySQL)	1	0.00	\$0.00
OpenProject	1	0.00	\$0.00
Paquete de Ofimática Libre Office	1	0.00	\$0.00
JADE	1	0.00	\$0.00
SUBTOTAL			\$ 0.00

- **Recursos de Comunicación**

El uso de internet y la adquisición del dominio, son parte de este recurso, que fueron imprescindibles para poder realizar las consultas. En la tabla 12 se detalla el costo por cada uno:

TABLA XII. RECURSOS DE COMUNICACIÓN.

RECURSOS DE COMUNICACIÓN	
RUBRO	COSTO
Internet	\$ 240.00
Llamadas Telefónicas (fijo, móvil)	\$ 40.00
SUBTOTAL:	\$ 280.00

- **Resumen del Presupuesto**

La tabla 13 se resume el balance del coste total del proyecto de tesis, para ello se suman los costes obtenidos totalmente.

TABLA XIII. COSTO TOTAL.

COSTO TOTAL	
RECURSOS	COSTOS
TALENTO HUMANOS	\$ 10240,00
BIENES	\$ 649.00
SERVICIO	\$ 280.00
IMPREVISTO	\$ 1116,90
TOTAL:	\$12285,90

El proyecto tiene un costo total de **\$\$12315,60**.

h. Conclusiones

- El Framework JADE facilito el desarrollo de los agentes, ya que este permite la ejecución múltiples agentes y de manera simultánea a través del modelo de comportamiento, así mismo establece un modelo estándar de comunicaciones para agentes a través de FIPA-ACL, haciendo que el transporte de mensajes sea de una manera eficiente y no presente errores en el mismo.
- El algoritmo permitió trazar la ruta utilizando los nodos establecidos basándose en los eventos producidos por agentes.
- El sistema multiagente resulta de gran ayuda de acuerdo a las pruebas realizadas para utilizarlo en la ciudad de Loja o en cualquier otra ciudad ya que nos ayudara a descongestionar el tráfico vehicular existente trazando rutas alternas en base a la ubicación de los semáforos para evitar accidentes o congestionamientos producidos en alguna intersección.

i. Recomendaciones

- Implementar los semáforos inteligentes e integrarlo con el Framework Jade en la ciudad de Loja para que se puedan comunicar entre sí, con el objetivo de poder tener un tráfico más organizado.
- Mejorar el algoritmo y se tracen rutas extensas cuando los puntos de origen y fin se encuentran distantes, tendríamos que crear agentes imaginarios y ubicarlos en todas las intersecciones creando una matriz de agentes la cual va a permitir realizar el trazo de la ruta con mayor éxito.
- Diseñar un mapa de la ciudad, estableciendo el lugar exacto de los semáforos, el sentido de las calles y permitir al usuario trazar su ruta desde dispositivos móviles.

j. Bibliografía

- [1] L. Hora, «La Hora,» La Hora, 12 Diciembre 2014. [En línea]. Available: http://www.lahora.com.ec/index.php/noticias/show/1101760764/-1/El_parque_automotor_de_Loja_crece_aceleradamente_.html. [Último acceso: 15 Abril 2015].
- [2] L. Hora, «La Hora,» 14 Enero 2014. [En línea]. Available: http://www.lahora.com.ec/index.php/noticias/show/1101450992/-1/Inicia_plan_para_colocar_sem%C3%A1foros_inteligentes_.html. [Último acceso: 30 Marzo 2015].
- [3] Diario la Hora, «La Hora,» 14 1 2013. [En línea]. Available: http://www.lahora.com.ec/index.php/noticias/show/1101450992/-1/Inicia_plan_para_colocar_sem%C3%A1foros_inteligentes_.html. [Último acceso: 4 Junio 2014].
- [4] R. J. Morales Linares y J. J. González Sánchez, «Universidad Rafael Urdaneta,» Julio 2013. [En línea]. Available: <http://200.35.84.131/portal/bases/marc/texto/2501-13-06247.pdf>. [Último acceso: 4 Junio 2014].
- [5] El mercurio, «El Mercurio,» 2 Octubre 2012. [En línea]. Available: <http://www.elmercurio.com.ec/351163-el-trafico-de-loja-preocupa/#.U462UvI5OE8>. [Último acceso: 4 Junio 2014].

- [6] Diario Cronica, «Cronica,» 24 1 2014. [En línea]. Available: <http://cronica.com.ec/index.php/loja/item/71764-16-intersecciones-tendr%C3%A1n-sem%C3%A1foros>. [Último acceso: 4 Junio 2014].
- [7] Cronica, «Cronica,» 16 Abril 2014. [En línea]. Available: <http://www.cronica.com.ec/index.php/opinion/item/73798->. [Último acceso: 4 Junio 2014].
- [8] M. Martínez. [En línea]. Available: http://www.jeuazarru.com/docs/semaforos_inteligentes.pdf. [Último acceso: 3 Junio 2014].
- [9] sistemasinformación2010, «trabajosi,» 17 Octubre 2010. [En línea]. Available: <http://trabajosi.wikispaces.com/SEMAFOROS+INTELIGENTES>. [Último acceso: 4 Junio 2014].
- [10] Proximidad, «Proximidad,» [En línea]. Available: <http://www.tarjetashid-mifare-rfid.com/que-es-tarjeta-proximidad-rfid.html>. [Último acceso: 6 Junio 2014].
- [11] A. Monteserin, 18 Noviembre 2013. [En línea]. Available: <http://www.exa.unicen.edu.ar/catedras/tmultiag/Clase1.pdf>. [Último acceso: 8 Junio 2014].
- [12] Universidad Veracruzana, «Universidad Veracruzana,» 2013. [En línea]. Available: <http://www.uv.mx/aguerra/documents/2013-ia2-01.pdf>. [Último acceso: 8 Junio 2014].

- [13] Fipa, «Fipa,» Fipa, [En línea]. Available: <http://www.fipa.org/subgroups/ROFS-SG-docs/2007-TAAS-specifying-MAS.pdf>. [Último acceso: 9 Junio 2014].
- [14] UPV, «Universidad Politecnica de Valencia.,» [En línea]. Available: <http://www.upv.es/sma/teoria/sma/Comunicaci%F3n%20entre%20agentes.pdf>. [Último acceso: 9 Junio 2014].
- [15] Universidad Politécnica de Madrid, «UPM,» [En línea]. Available: <http://www.sia.eui.upm.es/isa/lib/exe/fetch.php?media=asignaturas:acl.pdf>. [Último acceso: 9 Junio 2014].
- [16] Universidad Politécnica de Valencia, «upv,» [En línea]. Available: http://www.upv.es/sma/teoria/teoria_ag/Curso_BDI.pdf. [Último acceso: 9 Junio 2014].
- [17] Procesos de software, «Wikispaces,» [En línea]. Available: <http://procesosdesoftware.wikispaces.com/METODOLOGIAS+PARA+DESARROLLO+DE+SOFTWARE>. [Último acceso: 9 Junio 2014].
- [18] UDO, «Universidad del Oriente,» 11 Julio 2012. [En línea]. Available: http://wiki.monagas.udo.edu.ve/index.php/Metodolog%C3%ADas_para_el_desarrollo_de_software. [Último acceso: 9 Junio 2014].
- [19] H. Paz, «Desarrollo de un sistema multiagente mediante una arquitectura híbrida aplicado a dispositivos móviles: caso de estudio : citas médicas odontológicas,» Pachuca de Soto, Hgo.: UAEH, Instituto de Ciencias Básicas e Ingeniería, 2012, 2012.

- [20] N. Muñoz, C. Cobos, W. Rivera, J. López y M. Mendoza, «Uso de la metodología GAIA para modelar el,» Antioquia, 2010.
- [21] Jade, «Jade,» [En línea]. Available: www.jade.tilab.com. [Último acceso: 5 Enero 2015].
- [22] FIPA, «FIPA,» [En línea]. Available: <http://www.fipa.org/repository/aclspecs.html>. [Último acceso: 5 Enero 2015].
- [23] D. Lopez, S. Abraham Garcia y M. Gomez, «<http://decsai.ugr.es/~jfv/ed1/tedi/cdrom/docs/grafos.htm>,» [En línea]. Available: <http://decsai.ugr.es/~jfv/ed1/tedi/cdrom/docs/grafos.htm>. [Último acceso: 9 Junio 2014].
- [24] R. Lopez, E. Quevedo y A. Asencio, «<http://ulpgc.es/>,» uplgc, [En línea]. Available: http://www.iuma.ulpgc.es/users/jmiranda/docencia/programacion/Tema9_ne.pdf. [Último acceso: 8 Junio 2014].

k. Anexos

ANEXO 1. Informe de la Ubicación de los Semáforos.

GOBIERNO AUTÓNOMO DESCENTRALIZADO MUNICIPAL DE LOJA

UNIDAD MUNICIPAL DE TRANSPORTE TERRESTRE, TRÁNSITO Y SEGURIDAD VIAL

INFORME DE LA UBICACIÓN DE SEMÁFOROS 006 - ERMP – 2014

PARA: Ing. Tania Vásquez Romero

TÉCNICA DE LA UMTTTSV

DE: Sr. Édgar Martínez Palacios

FUNCIONARIO DE LA UMTTTSV

FECHA: 15-08-2014

Por medio del presente me permito informar a usted, la ubicación y el tipo de luz que poseen los semáforos de la ciudad de Loja, debiendo indicar que hay controladores por intersecciones, de acuerdo al siguiente detalle:

TABLA XIV. LUZ INCANDESCENTE.

Nº	UBICACIÓN CALLES /	TIPO DE LUZ	GRÁFICO	OBSERVACIONES
001	Calle Chile, Av. Universitaria y Av. Manuel Agustín Aguirre	Luz Incandescente		2 Intersecciones
002	Calle Celica y Av. Universitaria / Av. Manuel Agustín Aguirre y Calle Brasil	Luz Incandescente		2 Intersecciones
003	Calle Cariamanga y 18 de Noviembre	Luz Incandescente		1 Intersección

004	Calle Juan José Peña y Mercadillo	Luz Incandescente		1 Intersección
005	Calle Juan José Peña y Calle Rocafuerte	Luz Incandescente		1 Intersección
006	Calle Guaranda y Av. Emiliano O. / Av. Orillas del Zamora y Alderete	Luz Incandescente		1 Controla para 2 intersecciones
007	Av. Orillas del Zamora y Av. Juan de Salinas / Av, Emiliano Ortega y Av. Juan de S. / Calle Bernardo v. y Av. Juan de Salinas. / Calle Bolívar y Av. Juan de S.	Luz Incandescente		Controla 1 para 4 Intersecciones
008	Calle 10 de Agosto entre Av. Emiliano Ortega y Av. Orillas del Zamora	Luz Incandescente		2 Intersecciones
009	Calle Miguel Riofrío entre Av. Orillas del Zamora y Av. Emiliano Ortega	Luz Incandescente		2 Intersecciones
010	Calle 24 de Mayo entre Av. Emiliano Ortega y Av. Orillas del Zamora	Luz Incandescente		2 Intersecciones

Nº	UBICACIÓN / CALLES	TIPO DE LUZ	GRÁFICO	OBSERVACIONES
011	Av. Zoilo Rodríguez y 24 de Mayo, entre la Calle Paris y Atenas	Luz Incandescente	Pileta	3 Intersecciones
012	Calle Juan de Velasco entre Av. Santiago de las Montañas y calle Esmeraldas	Luz Incandescente		4 Intersecciones
013	Pasaje Santiago intersección Av. Orillas del Zamora	Luz Incandescente		2 Intersecciones
014	Calle Isidro Ayora entre Av. Isidro Ayora y Av.	Luz Incandescente		3 Intersecciones

	Salvador Bustamante Celi			
015	Av. Salvador Bustamante Celi y Agustín Carrión Palacios	Luz Incandescente		1 Intersección
016	Av. Guayaquil y Salvador Bustamante Celi	Luz Incandescente		1 Intersección
017	Av. Guayaquil entre Av. Nueva Loja y Av. Orillas del Zamora	Luz Incandescente		2 Intersecciones
018	Calle Gran Colombia entre Guaranda	Luz Incandescente		1 Intersección
019	Calle 8 de Diciembre entre Belisario Moreno	Luz Incandescente		1 Intersección
020	Calle Ibarra entre Av. Cuxibamba	Luz Incandescente		2 Intersecciones

Nº	UBICACIÓN CALLES /	TIPO DE LUZ	GRÁFICO	OBSERVACIONES
021	Calle Guaranda entre Av. Cuxibamba	Luz Incandescente		2 Intersecciones
022	Av. Manuel Agustín Aguirre y Av. Cuxibamba	Luz Incandescente		2 Intersecciones
023	Calle 10 de Agosto entre Ramón Pinto	Luz Incandescente		1 Intersección
024	Calle Miguel Riofrío entre calle Ramón Pinto	Luz Incandescente		1 Intersección
025	Calle Mercadillo, Ramón Pinto y Lauro Guerrero	Luz Incandescente		2 Intersecciones
026	Av. Pio Jaramillo y calle Teniente Maximiliano Rodríguez	Luz Incandescente		2 Intersecciones
027	Av. Pio Jaramillo y calle Cuba	Luz Incandescente		2 Intersecciones
028	Av. Pio Jaramillo y calle Chile	Luz Incandescente		2 Intersecciones

029	Av. Pio Jaramillo y calle Brasil	Luz Incandescente		2 Intersecciones
030	Av. Pio Jaramillo y calle España	Luz Incandescente		2 Intersecciones
Nº	UBICACIÓN CALLES /	TIPO DE LUZ	GRÁFICO	OBSERVACIONES
031	Calle Venezuela y José María Peña	Luz Incandescente		1 Intersección
032	Calle Rocafuerte y Lauro Guerrero	Luz Incandescente		1 Intersección
033	Calle 18 de Noviembre y Gobernación de Mainas	Luz Incandescente		2 Intersecciones
034	Av. Eduardo Kigman y calle Ángel Benigno Valarezo	Luz Incandescente	Rosales	2 Intersecciones
035	Av. Eduardo Kigman y calle Gonzanamá	Luz Incandescente	Cabo Minacho	2 Intersecciones
036	Av. Eduardo Kigman y calle Amaluzá	Luz Incandescente		2 Intersecciones
037	Av. Manuel Benjamín Carrión y calle Pablo Palacio	Luz Incandescente		2 Intersecciones
038	Calle Argentina y Bolivia	Luz Incandescente		2 Intersecciones
039	Calle Bernardo Valdivieso y Lourdes	Luz Incandescente		1 Intersección
040	Calle Bernardo Valdivieso y Mercadillo	Luz Incandescente		1 Intersección

Nº	UBICACIÓN CALLES /	TIPO DE LUZ	GRÁFICO	OBSERVACIONES
041	Calle Bernardo Valdivieso y Azuay	Luz Incandescente		1 Intersección
042	Calle Bernardo Valdivieso y Miguel Riofrío	Luz Incandescente		1 Intersección

043	Calle Bernardo Valdivieso y Rocafuerte	Luz Incandescente		1 Intersección
044	Calle Bernardo Valdivieso y 10 de Agosto	Luz Incandescente		1 Intersección
045	Calle Bernardo Valdivieso y José Antonio Eguiguren	Luz Incandescente		1 Intersección
046	Calle Bolívar e Imbabura	Luz Incandescente		1 Intersección
047	Calle Bolívar y Colón	Luz Incandescente		2 Intersección
048	Calle Bolívar y José Antonio Eguiguren	Luz Incandescente		1 Intersección
049	Calle Bolívar y 10 de Agosto	Luz Incandescente		1 Intersección
050	Calle Bolívar y Rocafuerte	Luz Incandescente		1 Intersección

Nº	UBICACIÓN CALLES /	TIPO DE LUZ	GRÁFICO	OBSERVACIONES
051	Calle Bolívar y Miguel Riofrío	Luz Incandescente		1 Intersección
052	Calle Bolívar y Mercadillo	Luz Incandescente		1 Intersección
053	Calle Bolívar y Lourdes	Luz Incandescente		1 Intersección
054	Calle Bolívar y Cariamanga	Luz Incandescente		1 Intersección
055	Calle Sucre y Cariamanga	Luz Incandescente		1 Intersección
056	Calle Olmedo y José Antonio Eguiguren	Luz Incandescente		1 Intersección
057	Calle Olmedo y Rocafuerte	Luz Incandescente		1 Intersección
058	Calle Olmedo y Miguel Riofrío	Luz Incandescente		1 Intersección

059	Calle Olmedo y Mercadillo	Luz Incandescente		1 Intersección
060	Calle Olmedo y Lourdes	Luz Incandescente		1 Intersección

Nº	UBICACIÓN CALLES /	TIPO DE LUZ	GRÁFICO	OBSERVACIONES
061	Calle 24 de Mayo y Rocafuerte	Luz Incandescente		1 Intersección
062	Calle 24 de Mayo y Azuay	Luz Incandescente		1 Intersección
063	Calle 24 de Mayo y Lourdes	Luz Incandescente		1 Intersección
064	Calle Sucre y Quito	Luz Incandescente		1 Intersección

TABLA XV. LUZ LED.

Nº	UBICACIÓN CALLES /	TIPO DE LUZ	GRÁFICO	OBSERVACIONES
065	Calle Alonso de Mercadillo entre Av. Iberoamérica y Av. Universitaria	Luz Led		1 Intersección
066	Calle Rocafuerte entre Av. Manuel Agustín Aguirre y Av. Universitaria	Luz Led		1 Intersección
067	Calle 10 de Agosto entre Av. Manuel Agustín y Av. Universitaria	Luz Led		1 Intersección
068	Calle José Antonio Eguiguren, av. Universitaria y av. Manuel Agustín Aguirre	Luz Led		1 Intersección

069	Calle José Félix V., Universitaria y av. Manuel Agustín Aguirre	Luz Led		1 Intersección
070	Av. Juan de Salinas, av. Universitaria y av. Manuel Agustín Aguirre	Luz Led		1 Intersección
071	Calle Juan J. Samaniego y Av. Manuel Agustín A., Av Universitaria e Imbabura	Luz Led		1 Intersección
072	Calle Colón, Av. Manuel Agustín A. y Av. Universitaria	Luz Led		1 Intersección
073	Calle Miguel Riofrío, Av. Manuel Agustín A. y Av. Universitaria	Luz Led		1 Intersección
074	Calle Azuay, Av. Manuel Agustín A. y Av. Universitaria	Luz Led		1 Intersección

Nº	UBICACIÓN CALLES /	TIPO DE LUZ	GRÁFICO	OBSERVACIONES
075	Calle Sucre y Lourdes	Luz Led		
076	Calle Sucre y Mercadillo	Luz Led		2 Intersecciones, esta intersección controla de la Sucre y Lourdes
077	Calle Sucre y Azuay	Luz Led		2 Intersecciones, esta intersección controla de la Sucre y Miguel y Miguel Riofrío
078	Calle Sucre y Miguel Riofrío	Luz Led		
079	Calle Sucre y Rocafuerte	Luz Led		1 Intersección
080	Calle Sucre y 10 de Agosto	Luz Led		1 Intersección
081	Calle Sucre y José Antonio Eguiguren	Luz Led		1 Intersección

082	Calle Sucre y Colón	Luz Led		1 Intersección
083	Calle Sucre e Imbabura	Luz Led		1 Intersección
084	Calle Sucre y Av. Juan de Salinas	Luz Led		1 Intersección

Nº	UBICACIÓN / CALLES	TIPO DE LUZ	GRÁFICO	OBSERVACIONES
085	Calle 18 de Noviembre y Av. Juan de Salinas	Luz Led		1 Intersección
086	Calle 18 de Noviembre e Imbabura	Luz Led		2 Intersecciones, esta intersección controla la calle 18 de Noviembre y Colón
087	Calle 18 de Noviembre y Colón	Luz Led		
088	Calle 18 de Noviembre y José Antonio Eguiguren	Luz Led		2 Intersecciones, esta intersección controla la calle 18 de Noviembre y de Agosto
089	Calle 18 de Noviembre y 10 de Agosto	Luz Led		
090	Calle 18 de Noviembre y Miguel Riofrío	Luz Led		
091	Calle 18 de Noviembre y Azuay	Luz Led		2 Intersecciones, esta intersección controla la calle 18 de Noviembre y Miguel Riofrío
092	Calle 18 de Noviembre y Mercadillo	Luz Led		1 Intersección

ANEXO 2. Artículo Científico.

Desarrollo y simulación de un sistema multiagente para la comunicación de semáforos para encontrar la ruta óptima mediante grafos.

Robert Iván Loarte Caraguay^a, Bolivar Rolando Quizhpe Vazques^a.

^a Carrera de ingeniería de Sistemas, Universidad Nacional de Loja
Loja – Ecuador
robert.loarte@gmail.com, bologv.88@gmail.com

Resumen. En el presente artículo trata sobre el desarrollo y simulación de un sistema multiagentes que serán representados por semáforos con el objetivo de que estos se comuniquen a través de un lenguaje establecido y logren tomar decisiones de manera autónoma, esto desarrollado a través de una arquitectura denominada BDI (Beliefs: Creencias, Desires: Deseos, Intentions: Intenciones) y a través de JADE (Framework para el desarrollo de agentes en Java) con FIPA-ACL que soporta la coordinación de múltiples agentes y proporciona un lenguaje estándar de comunicación entre ellos. Los semáforos se comunican cuando se haya producido un evento, por ejemplo: Accidente, aglomeración y cuando exista un nuevo agente, se comunican los semáforos más cercanos al semáforo donde se produjo el evento. Cada vez que exista un accidente o aglomeración, estos semáforos se marcaran en el mapa, dando la posibilidad al usuario de evitar esos semáforos y trazándose una ruta hacia su lugar de destino.

Palabras Clave: Sistemas Multiagentes, Jade, FIPA-ACL, Ruta Optima.

1 Introducción

El incremento [1] del parque automotriz que se produce cada año en el país, hace que la parte céntrica de las ciudades del país se produzca congestionamientos y con ellos posibles accidentes, al mismo tiempo la contaminación que produce al medio ambiente.

El desarrollo de sistemas inteligentes para el control de tráfico automotriz, permite controlar el tráfico más eficaz y eficientemente, se controla en tiempo real detectando accidentes, verificando si existe aglomeración de vehículos o de personas y ayudando a los conductores a elegir rutas alternas para poder llegar a su destino evitando contratiempos, permitiendo el ahorro de combustible, control de tiempo por parte del conductor y mayor fluidez en el tráfico vehicular y peatonal.

En la actualidad los semáforos son sistemas temporizados para cambiar de un estado a otro y siguen un patrón de secuencia fija, ya que carece de inteligencia para tomar decisiones; esto presenta una gran desventaja en las horas pico. Para ayudar a solventar el tráfico de manera más eficaz se propone utilizar semáforos inteligentes ya que de acuerdo al grado de congestión que presente alguna vía de una intersección prolongue un poco más el paso de los automóviles para así descongestionar las principales arterias viales.

Actualmente los semáforos en la ciudad en Loja [2], la mayoría de ellos son temporizados y algunos mecánicos, por esa razón se sincronizan en un tiempo específico para cambiar de estado “Rojo, verde, amarillo”.

Datos otorgados por la unidad municipal de transporte terrestre de la ciudad Loja a diciembre del 2014 actualmente existen 117 semáforos que se distribuyen alrededor de la ciudad de Loja.

38 semáforos con tecnología LED.

79 semáforos Incandescentes.

Al existir semáforos mecánicos y semáforos temporizados mezclados, estos no pueden sincronizarse, ni tampoco pueden comunicarse entre sí, imposibilitando que estos semáforos se adapten a las condiciones de tráfico en tiempo real, dando en ocasiones prioridad a calles vacías sin tráfico que calles que se encuentran saturadas.

El objetivo primordial de este proyecto es crear una arquitectura multiagente representado por semáforos que permitan comunicarse entre sí, desarrollado a través de un Framework denominado JADE y con estándares de comunicación FIPA-ACL, simulando accidentes o aglomeraciones los semáforos podrán tomar decisiones por si mismos e informar a sus semáforos más cercanos sobre un evento que se ha producido en su intersección.

El sistema multiagente también está diseñado para que el usuario evite pasar por los accidentes o aglomeraciones que existan, el usuario elije el punto de origen y el destino y se trazara una ruta evitando dicho eventos, ayudando a descongestionar las vías y haciendo el tráfico más fluido.

2 Características de las herramientas de software.

Para desarrollar el sistema multiagente para la comunicación de semáforos se utilizó el frameowk JADE desarrollado en Java.

JADE [3] (Java Agent DEvelopment Framework): Es una plataforma de código abierto para aplicaciones de agentes basados en peer-to-peer.

JADE está desarrollado en java, esto simplifica significativamente el desarrollo de sistema multiagetes y que cumple con las especificaciones FIPA [4]. Un sistema basado en JADE puede ser distribuido a través de una red de computadoras independientemente del sistema operativo y se puede controlar a través de una interfaz de usuario remotamente. JADE se implementa por completo en Lenguaje java y la exigencia mínima es la versión 5 de java en adelante.

La arquitectura basada para el desarrollo del sistema Multiagemte es la BDI (Beliefs, Desires, Intentions), en esta arquitectura el agente es visto como un agente autónomo racional, capaz de tomar decisiones por si mismos a través de las creencias y deseos almacenados en una base de datos e intenciones.

- Las creencias son información que contiene al agente de su entorno, en este caso el entorno puede ser una aglomeración, accidente o cuando se crea un nuevo agente.
- Los agentes poseen un conjunto de deseos, el cual los agentes los toma como prioridades o como objetivos principales, por ejemplo: Informar que se ha producido un accidente a los semáforos más cercanos o informar de qué existe un nuevo agente.
- Las intenciones son los deseos elegidos y dicho agente informara a los agentes más cercanos sobre el deseo elegido que se encuentra almacenado en una base de datos.

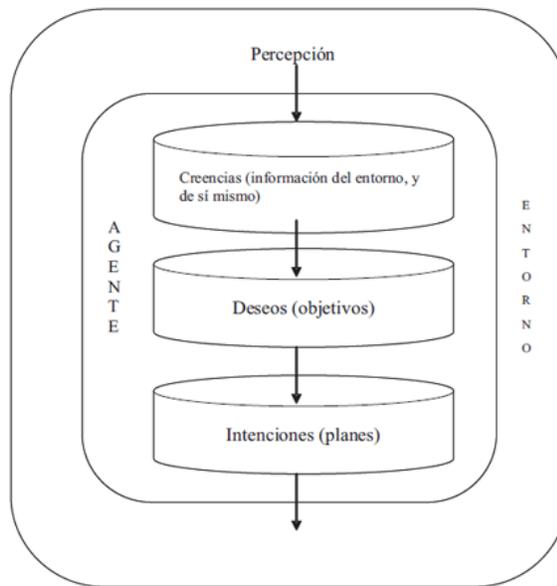


Figura 1. Arquitectura BDI básica.

FIPA-ACL permite que los agentes se comuniquen a través del intercambio de mensajes, que representan actos de habla, codificados en un lenguaje de comunicación de agentes, así mismo FIPA-ACL brinda el soporte para el transporte de mensajes y servicio de directorios.

El servicio de transporte de mensajes que provee FIPA-ACL es confiable ya que dicho mensaje llega sin alteraciones a su destino, así mismo fiable ya que dicho mensaje se recibe tal y como se envía.

La condición que se necesita para que exista comunicación entre agentes es que dicho agentes deben tener un ID único que les diferencie de todos los demás y un localizador que describen el tipo de transporte y la dirección de transporte específico para cada agente.

En la figura 2 se puede observar un extracto de comunicación de agentes con sus respectivos ID's y Localizadores.

```
(request
  :sender (agent-identifier
    :name Mind@localap)
  :receiver (agent-identifier
    :name Body@localap)
  :content
    "Tactic=
    IncreaseStudentSelfEfficacy"
  :language "BodyAgent"
  :protocol fipa-request
)
```

Figura 2. Comunicación de agentes.

El sistema puede trazar una ruta evitando los accidentes/aglomeraciones que existan en la parte céntrica de la ciudad de Loja, el punto de origen y destino son dados por el usuario en este caso el conductor. Para lograr trazar una ruta, se diseñó un algoritmo basándose en grafos donde cada nodo es un semáforo.

3 Metodología de desarrollo de software.

Para el desarrollo del sistema Multiagente se hizo uso de la metodología **GAIA** [5]

GAIA es una metodología enfocada al análisis y diseño de sistemas de software basados en agentes inteligentes, define al sistema multiagentes como un conjuntos de roles y organizaciones que posee un agente.

- Fase de Análisis. En esta fase se define los roles principales que tendrá cada agentes y establecer las relaciones entre cada agente. Los modelos de interacción define los protocolos a usar, así como los agentes que iniciaran una iteración y los receptores y el procesamiento de información.
- Fase de diseño: Es el diseño concreto donde existen los modelos de agentes y el modelo de comunicación representados por el diagrama de secuencia.

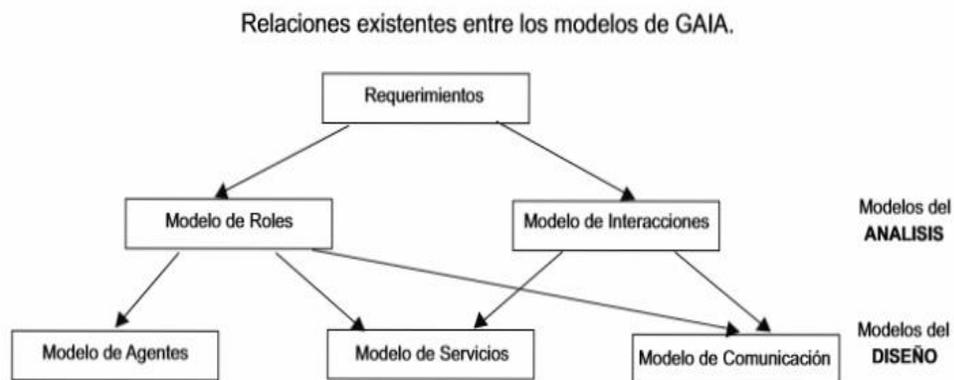


Figura 3. Metodología GAIA.

4 Resultados.

Para el desarrollo del siguiente sistema multiagente, se ha utilizados diversas herramientas, dando como resultado el sistema de comunicación de semáforos y trazado de ruta para evitar accidentes que se describen a continuación.

Análisis, diseño y desarrollo de agentes inteligentes y su comunicación entre sí.

A continuación se detalla el proceso para el desarrollo de los agentes inteligentes y su comunicación a partir del análisis y diseño usando la metodología GAIA.

4.1 Análisis del sistema:

De acuerdo a la metodología GAIA en la fase de análisis se definen los roles del sistema y el modelo de interacciones.

Después de una análisis profundo se estableció las creencias y los deseos que poseerá cada agente y los roles que cumplirá cada uno y como interactuaran entre, teniendo en cuenta que los agentes son emisores y receptores.

- Creencias:
 - Hay comunicación.
 - No hay comunicación.
 - Hay accidente.
 - No hay accidente.
 - Nuevo agente.
 - No existe un agente.
 - Hay congestiónamiento.
 - No hay congestiónamiento.

- Deseos:
 - Informar accidentes.
 - Informar congestiónamiento.
 - Existe un nuevo agente.

4.1.1 Modelos de roles.

Roles de los agentes:

- Establecer comunicación entre agentes.
- Informar aglomeración.
- Informar accidente.
-

Tabla 1. Establecer comunicación

Esquema del Rol:	Establecer comunicación
Descripción.	Se encarga de establecer la comunicación entre los agentes existentes y nuevos agentes.
Actividades y protocolos.	. El nuevo agente envía mensajes a agentes cercanos. . Envío de mensajes entre agentes existentes. . Recepción de mensaje de disponibilidad del agente vecino.
Permisos:	Lectura: mensaje de disponibilidad. Cambio: Genera: respuesta si el agente está disponible.
Responsabilidades:	
Dinámicos:	

Seguridad.

Tabla N° 2 Informar congestionamiento.

Esquema del Rol: Informar congestionamiento.
Descripción. Se encarga de informar a los semáforos vecinos sobre un congestionamiento.
Actividades y protocolos. Establecer comunicación con los agentes vecinos más cercanos. Enviar información que ha existido un congestionamiento. Confirmar recepción de mensaje de ubicación
Permisos Lectura: Mensaje de información de ubicación del accidente. Cambio: Semáforo vecino cambia de estado (Rojo) Genera:
Responsabilidades: Dinámicos. Cambio el estado de los semáforos. Seguridad. Ninguno

Tabla N°3 Informar accidente

Esquema del Rol Informar accidente
Descripción. Informar a los semáforos vecinos sobre un accidente.
Actividades y protocolos. Establecer comunicación con los agentes vecinos más cercanos. Envía mensaje de información de accidente. Confirma recepción de mensaje de accidente.

Permisos: Lectura: mensaje de información de accidente Cambio: Genera:
Responsabilidades: Dinámicos. Seguridad.

4.1.2 Modelo de Interacciones

El modelo de interacciones define los protocolos a usar, cual es el objetivo de la interacción y especificar los objetivos del emisor y receptor.

Ejemplo:

. **Tabla N°4** Establecer la comunicación entre agentes

Protocolo	Establecer comunicación entre agentes.
Objetivo.	Se encarga de establecer la comunicación entre los agentes existentes y nuevos agentes.
Iniciador.	Agente emisor.
Receptor.	Agentes receptores.
Entradas.	Mensaje del agente emisor
Salidas.	Respuesta de los agentes receptores.
Procesamiento.	El agente iniciador envía un mensaje para saber si el agente se encuentra disponible, el receptor responde el mensaje para informar su disponibilidad.

Tabla N°5 Informar congestionamiento.

Protocolo	Informar congestionamiento.
Objetivo.	Se encarga de informar a los semáforos vecinos sobre un congestionamiento.
Iniciador.	Agente emisor.
Receptor.	Agentes receptores.
Entradas.	Los agentes receptores reciben información sobre un congestionamiento.
Salidas.	Agente cambia de estado bloqueando el acceso a los vehículos al lugar del congestionamiento.
Procesamiento.	El agente emisor, envía mensaje de información sobre congestionamiento a los agentes más cercanos, los agentes receptores envían mensaje de confirmación y cambian el estado del semáforo.

Tabla N°6 Informa accidente.

Protocolo	Informar accidente
Objetivo.	Se encarga de informar a los semáforos vecinos sobre un accidente.
Iniciador.	Agente emisor.
Receptor.	Agentes receptor.
Entradas.	Los agentes receptores reciben información sobre un accidente.
Salidas.	Agente cambia de estado bloqueando el acceso a los vehículos al lugar del accidente.
Procesamiento.	El agente emisor, envía mensaje de información sobre accidente a los agentes más cercanos, los agentes receptores envían mensaje de confirmación y cambian el estado del semáforo.

4.1.3 Diseño del sistema

Se diseñó el diagrama de clases, el modelo de agentes y el modelo de comunicaciones.

- Diagrama de clases.

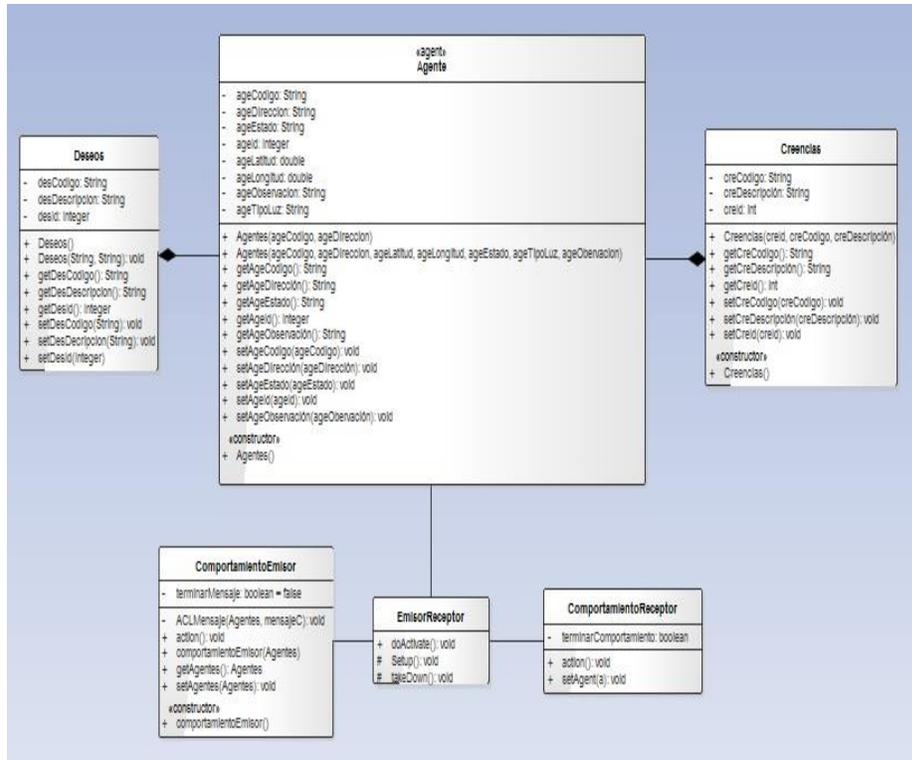


Figura 4. Diagrama de clases

- **Modelo de agentes:**

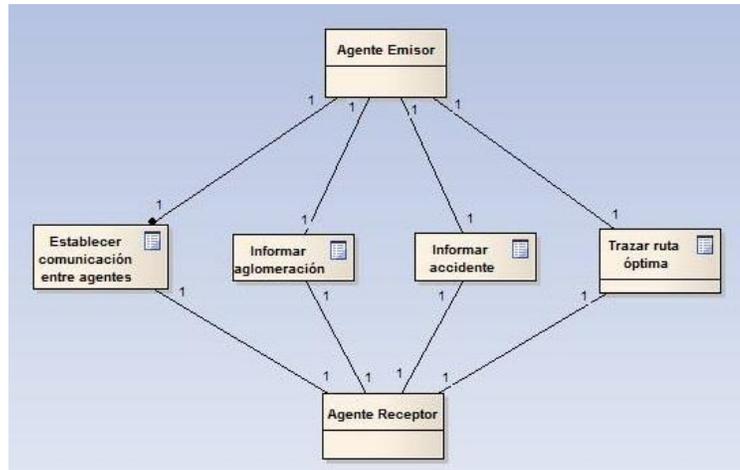


Figura 5. Modelado de agentes

- **Diagrama de comunicación.**

Se realiza el diagrama de comunicación, de los roles de cada agente.

1. Informar accidente.

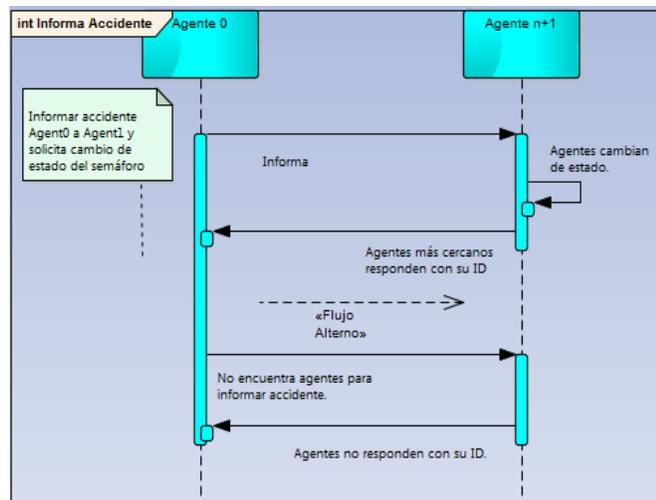


Figura 6. Informar accidente

2. Establecer comunicación de agentes.

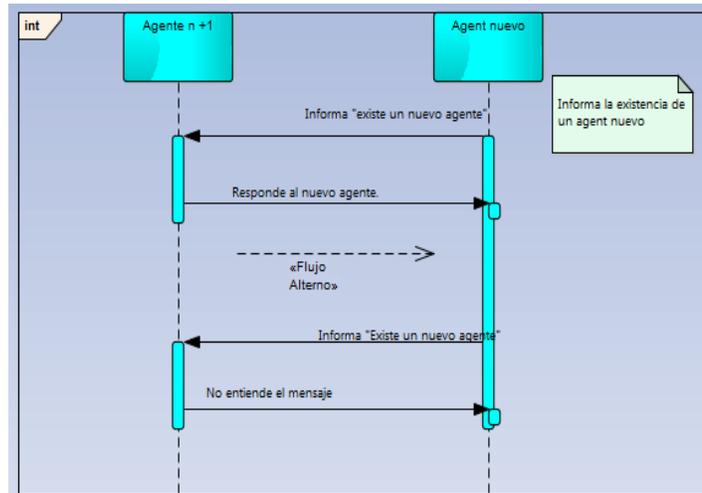


Figura 7. Establecer comunicación de agentes.

3. Informar congestiónamiento.

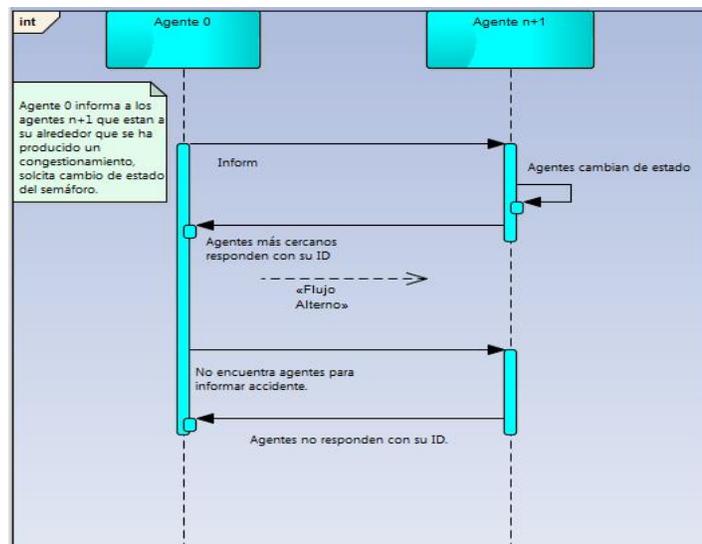


Figura 8. Informar congestiónamiento.

Terminado con el análisis y diseño se programan los agentes en el lenguaje de programación Java, se utiliza el mapa de google maps de la ciudad de Loja y se establecen los marcadores en el mapa representados por semáforos en cada intersección, donde existan semáforos funcionando en la actualidad.



Figura 9. Vista semáforos

4.2 Comunicación de semáforos.

Para producir el intercambio de mensajes se selecciona un semáforo cualquiera, una vez seleccionado se simula un accidente o aglomeración, produciéndose automáticamente el intercambio de mensajes.

Ejemplo. Accidente simulado en el centro de la ciudad de Loja, calles Bernardo Valdivieso y Miguel Riofrio.

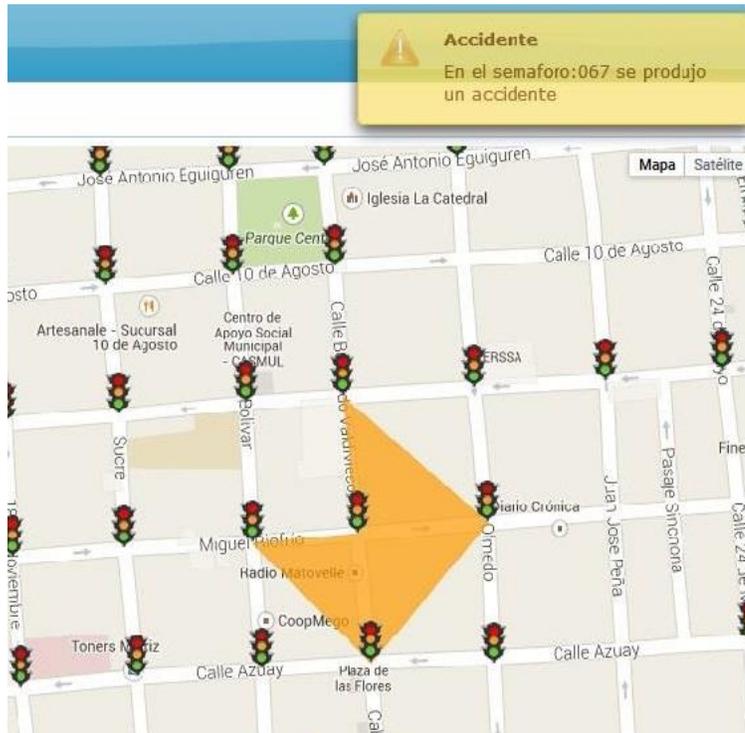


Figura 10. Crear accidente

Una vez que se simula el accidente, internamente se produce el intercambio de mensajes, el semáforo emisor envía el mensaje a sus semáforos más cercanos con el objetivo de que no ingrese tráfico y así evitar aglomeraciones o posibles accidentes.

```

Información: mensajes a enviar mensjes066
Información: mensajes a enviar mensjes069
Información: mensajes a enviar mensjes068
Información: mensajes a enviar mensjes055
Información: mensaje:(INFORM
:sender ( agent-identifier :name "067@192.168.0.105:1099/JADE"
:addresses (sequence http://Equipo:61881/acc ))
:receiver (set ( agent-identifier :name "066@192.168.0.105:1099/JADE" )
( agent-identifier :name "069@192.168.0.105:1099/JADE" )
( agent-identifier :name "068@192.168.0.105:1099/JADE" )
( agent-identifier :name "055@192.168.0.105:1099/JADE" ) )
:content "Informar accidente"
:language Español )
Información: Mensaje recibido
-----
Información: El agente 067 dice: Informar accidente
Información: Agente:066 responde
Información: mensaje:(INFORM
:sender ( agent-identifier :name "067@192.168.0.105:1099/JADE"
:addresses (sequence http://Equipo:61881/acc ))
:receiver (set ( agent-identifier :name "066@192.168.0.105:1099/JADE" )
( agent-identifier :name "069@192.168.0.105:1099/JADE" )
( agent-identifier :name "068@192.168.0.105:1099/JADE" )
( agent-identifier :name "055@192.168.0.105:1099/JADE" )
)
:content "Informar accidente"
:language Español )
Información: Mensaje recibido
-----

```

```

-----
Información: El agente 067 dice: Informar accidente
Información: Agente:068 responde
Información: mensaje:(INFORM
:sender ( agent-identifier :name "067@192.168.0.105:1099/JADE"
:addresses (sequence http://Equipo:61881/acc ))
:receiver (set ( agent-identifier :name "066@192.168.0.105:1099/JADE" )
( agent-identifier :name "069@192.168.0.105:1099/JADE" )
( agent-identifier :name "068@192.168.0.105:1099/JADE" )
( agent-identifier :name "055@192.168.0.105:1099/JADE" ) )
:content "Informar accidente"
:language Español )
Información: Mensaje recibido
-----
Información: El agente 067 dice: Informar accidente
Información: Agente:055 responde
Información: mensaje:(INFORM
:sender ( agent-identifier :name "067@192.168.0.105:1099/JADE"
:addresses (sequence http://Equipo:61881/acc ))
:receiver (set ( agent-identifier :name "066@192.168.0.105:1099/JADE" )
( agent-identifier :name "069@192.168.0.105:1099/JADE" )
( agent-identifier :name "068@192.168.0.105:1099/JADE" )
( agent-identifier :name "055@192.168.0.105:1099/JADE" )
)
:content "Informar accidente"
:language Español )
Información: Mensaje recibido

```

Figura 11.Mensajes accidente

4.3 Simulación de aglomeración.

Se simula aglomeración y al igual que accidente se elige el semáforo y esté actuando de emisor, informa a sus semáforos más cercanos a su intersección.

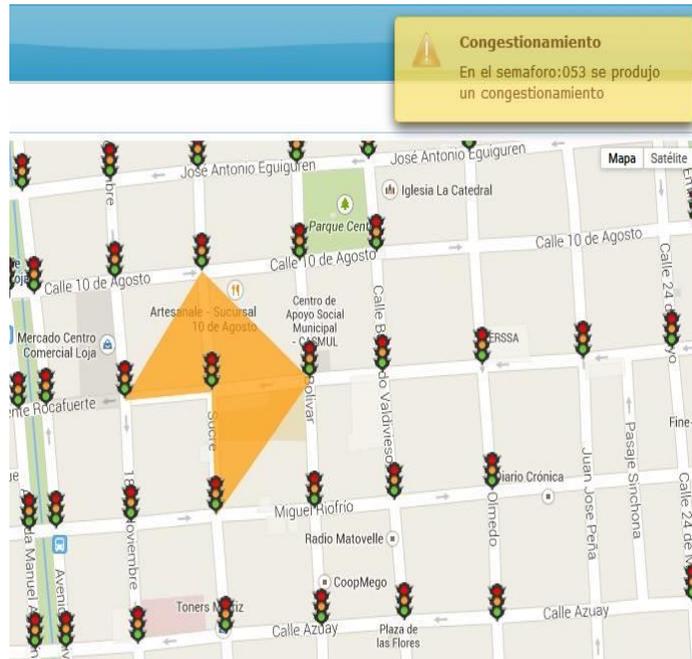


Figura 12. Congestionamiento

El resultado del intercambio de mensajes que se produce internamente es el siguiente.

```

Información: mensajes a enviar mensjes129
Información: mensajes a enviar mensjes042
Información: mensajes a enviar mensjes054
Información: mensajes a enviar mensjes065
Información: mensaje:(INFORM
:sender ( agent-identifier :name "053@192.168.0.105:1099/JADE"
:addresses (sequence http://Equipo:63344/acc ))
:receiver (set ( agent-identifier :name "129@192.168.0.105:1099/JADE" )
( agent-identifier :name "042@192.168.0.105:1099/JADE" )
( agent-identifier :name "054@192.168.0.105:1099/JADE" )
( agent-identifier :name "065@192.168.0.105:1099/JADE" ) )
:content "Informar congestionamiento"
:language Español )
Información: Mensaje recibido
-----
Información: El agente 053 dice: Informar congestionamiento
Información: Agente:054 responde
Información: mensaje:(INFORM
:sender ( agent-identifier :name "053@192.168.0.105:1099/JADE"
:addresses (sequence http://Equipo:63344/acc ))
:receiver (set ( agent-identifier :name "129@192.168.0.105:1099/JADE" )
( agent-identifier :name "042@192.168.0.105:1099/JADE" )
( agent-identifier :name "054@192.168.0.105:1099/JADE" )
( agent-identifier :name "065@192.168.0.105:1099/JADE" ) )
:content "Informar congestionamiento"
:language Español )
Información: Mensaje recibido
-----

```

Figura 13. Mensajes congestionamiento

4.4 Creación de un nuevo Agente.

Cuando se crea un nuevo semáforo, este informa a sus más cercanos su existencia y también se conoce su Latitud y Longitud.

Ejemplo.

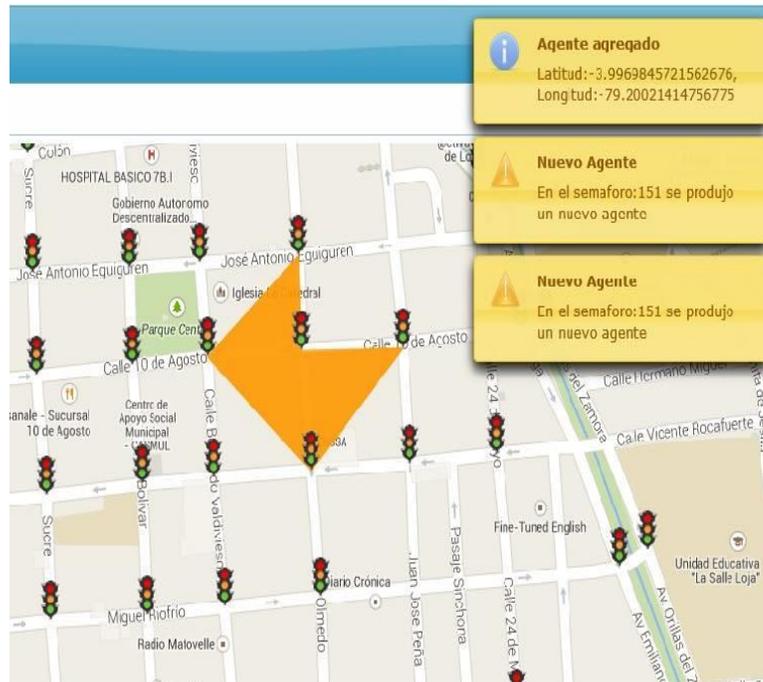


Figura14. Nuevo agente

El intercambio de mensajes se produce internamente y no es visible para un usuario común.

A continuación se visualiza el intercambio de mensajes cuando se crea un nuevo agente y es añadido al mapa de la ciudad de Loja.

```

Información: mensajes a enviar mensjes044
Información: mensajes a enviar mensjes056
Información: mensajes a enviar mensjes150
Información: El agente 151 dice: existe nuevo agente
Información: El agente 038 dice: deliberando y el Objeto es:
Información: mensaje:(INFORM
:sender ( agent-identifier :name "151@192.168.0.105:1099/JADE"
:addresses (sequence http://Equipo:64068/acc )
:receiver (set ( agent-identifier :name "038@192.168.0.105:1099/JADE" )
( agent-identifier :name "044@192.168.0.105:1099/JADE" )
(agent-identifier :name "056@192.168.0.105:1099/JADE" )
(agent-identifier :name "150@192.168.0.105:1099/JADE" ) )
:content "existe nuevo agente"
:language Español )
Información: Agente:038 responde
Información: Informar nuevo agente
Información: Mensaje recibido
-----
Información: El agente 151 dice: existe nuevo agente
Información: mensaje:(INFORM
:sender ( agent-identifier :name "151@192.168.0.105:1099/JADE"
:addresses (sequence http://Equipo:64068/acc )
:receiver (set ( agent-identifier :name "038@192.168.0.105:1099/JADE" )
( agent-identifier :name "044@192.168.0.105:1099/JADE" )
(agent-identifier :name "056@192.168.0.105:1099/JADE" )
(agent-identifier :name "150@192.168.0.105:1099/JADE" ) )
:content "existe nuevo agente"
:language Español )
Información: Agente:044 responde
Información: Informar nuevo agente

```

Figura 15. Mensajes nuevo agente

5 Diseño del grafo para evitar accidentes y aglomeración.

Para la simulación de la ruta que evite los accidentes o aglomeraciones que puedan existir se basó en el funcionamiento del algoritmo de dijkstra, donde cada intersección donde se encuentre el semáforo será tomada como un nodo, al que se le denominara nodo “padre” y cada nodo o semáforo siguiente se le denominara como nodo “hijo” que tendrá la misma dirección y sentido del nodo padre y el nodo hijo 2 que se encontrara en la vía transversal y será el próximo nodo, tomado desde el nodo padre. Para trazar la ruta entre dos puntos el sistema buscara el agente más cercano evitando el agente donde se produjo el accidente o congestión.

Ejemplo:

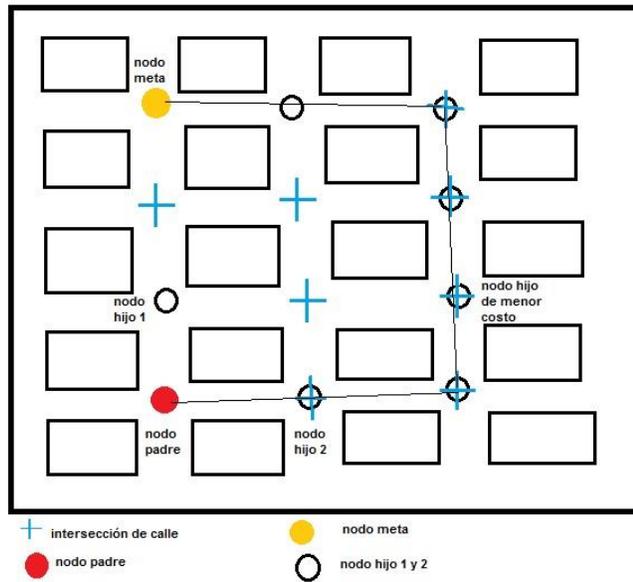


Figura 16. Grafo

Cada vez que se produzca el accidente o aglomeración, el usuario debe elegir el lugar de destino al que desea llegar, una vez determinado el lugar de destino, el sistema trazara la ruta y evitara los accidentes o aglomeraciones.

Como se podrá observar en la siguiente figura, el accidente se produjo en la calle Bolívar y Vicente Rocafuerte, el usuario selecciona el lugar de origen (A) y destino (B) y el sistema trazara la ruta, evitando dichos eventos.

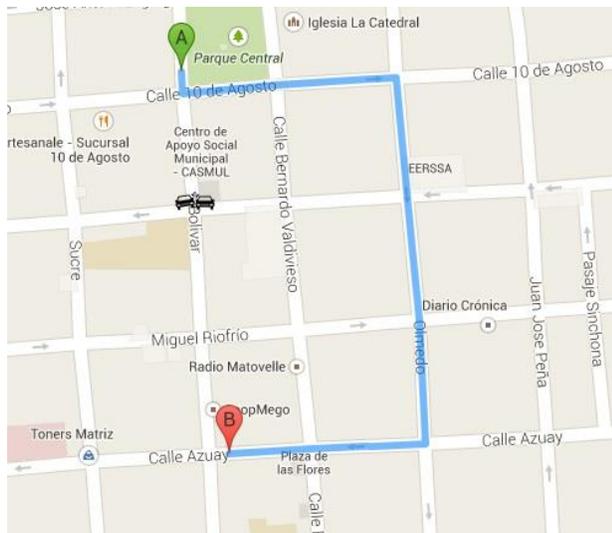


Figura 17. Simulación ruta optima

A continuación se simula accidentes en la parte céntrica de la ciudad de Loja, donde existe la mayor de cantidad de semáforos y el trazado de ruta.

Accidente simulado: Calle 10 de Agosto y Sucre

Congestionamiento: V. Rocafuerte y Bolívar.

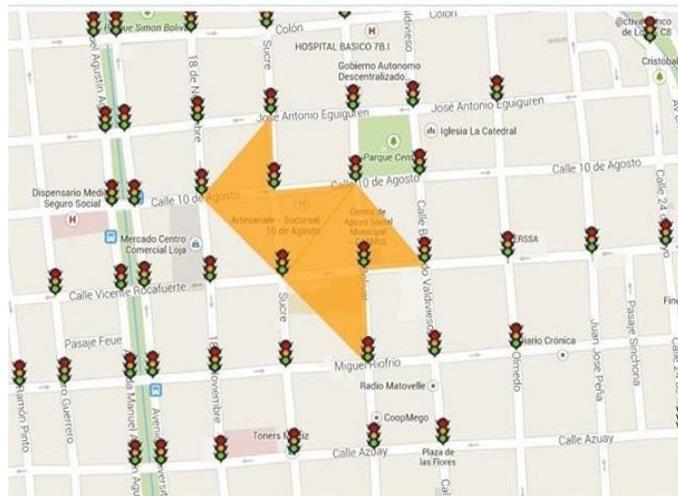


Figura 18. Pruebas 1

Lugar de origen: José Antonio Eguigure y 18 de noviembre

Lugar de destino: Juan José Peña y Vicente Rocafuerte.

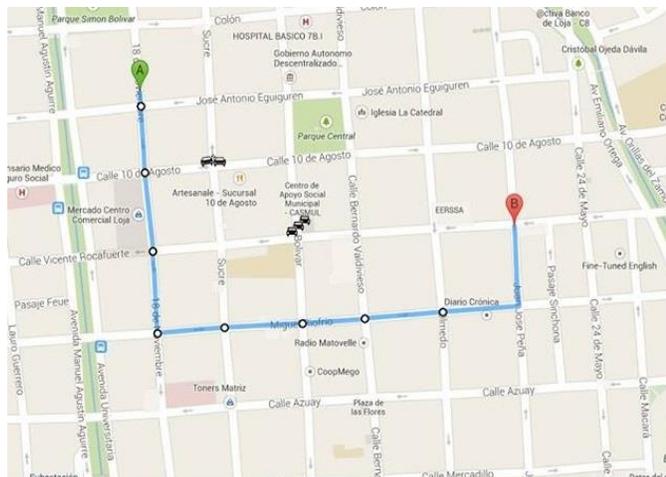


Figura 19. Pruebas 2

Accidentes: 18 de Noviembre y V. Rocafuerte, Bolívar y V. Rocafuerte, Bernardo Valdivieso y Miguel Riofrío, Sucre y Miguel Riofrío.

Aglomeraciones: Bolívar y Miguel Riofrío, Bolívar y Azuay.

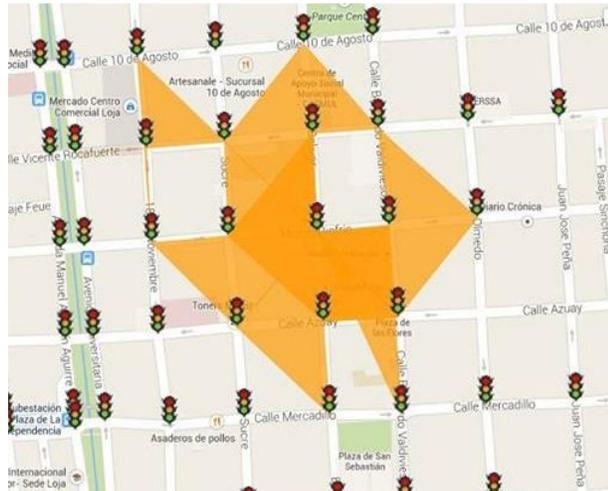


Figura 20. Pruebas 3

Lugar de origen: 10 de agosto y Bolívar

Lugar de destino: Calle Olmedo y Lourdes

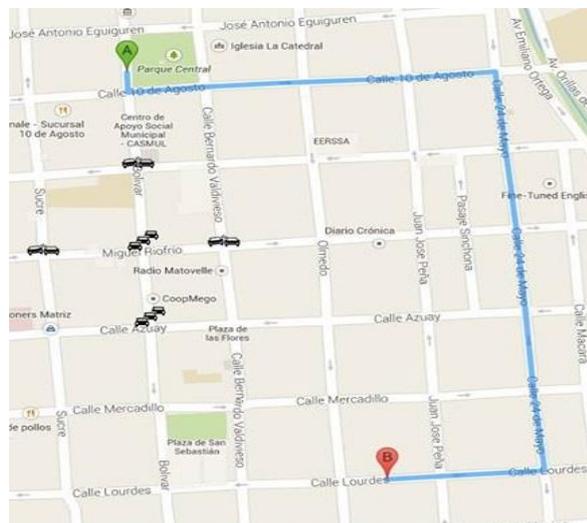


Figura 21. Pruebas 4

6 CONCLUSIONES.

Utilizar JADE para la construcción de agentes resulto de gran ayuda ya que nos facilitó el desarrollo de los agentes inteligentes para la comunicación de semáforos en tiempo real.

La metodología GAIA nos permitió obtener de manera eficaz los roles y responsabilidades de los agentes.

El sistema multiagente resulta de gran ayuda de acuerdo a las pruebas realizadas para utilizarlo en la ciudad de Loja ya que nos ayudara a descongestionar el tráfico vehicular existente trazando rutas alternas en base a la ubicación de los semáforos para evitar accidentes o congestionamientos producidos en alguna intersección.

7 REFERENCIAS

- [1] L. Hora, «La Hora,» La Hora, 12 12 2014. [En línea]. Available: http://www.lahora.com.ec/index.php/noticias/show/1101760764/-1/El_parque_automotor_de_Loja_crece_aceleradamente_.html. [Último acceso: 1 Abril 2015].
- [2] L. Hora, «La Hora,» 14 Enero 2013. [En línea]. Available: http://www.lahora.com.ec/index.php/noticias/show/1101450992/-1/Inicia_plan_para_colocar_sem%C3%A1foros_inteligentes_.html. [Último acceso: 30 Marzo 2015].
- [3] Tilap, «Jade Site,» [En línea]. Available: <http://jade.tilab.com/>. [Último acceso: 2 Abril 2015].
- [4] FIPA, «The Foundation For Intelligent Physical Agents,» [En línea]. Available: <http://www.fipa.org/>. [Último acceso: 6 Abril 2015].
- [5] M. Pérez López y A. F. Vergara, «Metodología GAIA,» Bogotá, 2012.

ANEXO 3. Certificación Traducción Summary

 **Washington**
ENGLISH INSTITUTE

WEIL - Nº 0000162

Más práctica por minuto... menor tiempo de aprendizaje.....

Yo, Yanina Quizhpe Espinosa, profesora del Centro Washington English;

Certifico:

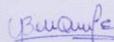
Que tengo el conocimiento y dominio de los idiomas español e inglés y que las traducciones de los siguientes:

RESUMEN del tema:

“DESARROLLO Y SIMULACIÓN DE UN SISTEMA MULTIAGENTE PARA LA COMUNICACIÓN DE SEMÁFOROS PARA ENCONTRAR LA RUTA OPTIMA MEDIANTE GRAFOS”.

para: QUIZHPE VASQUEZ ROLANDO BOLIVAR

es verdadero y correcto a mi mejor saber y entender.


Firmado en Loja a los veintitrés días del mes de abril de 2015.


WASHINGTON ENGLISH
DIRECCION ACADÉMICA
Formar Cita Lusa
Loja - Ecuador

24 de Mayo 11 - 20 y Azuay - 2573489 - 2579934