

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:
Ingeniera e Ingeniero de Sistemas**

**TEMA:
INTERFAZ GRÁFICA PARA LA VISUALIZACIÓN DE UNA RED DE
GRAFOS GEOREFERENCIADOS DE LAS PRINCIPALES CALLES DEL
CENTRO DE LA CIUDAD DE QUITO.**

**AUTORES:
ANDREA ESTEFANÍA HURTADO TERÁN
JONNY JAVIER LLUMIGUSÍN CHAMBA**

**TUTORA:
LINA PATRICIA ZAPATA MOLINA**

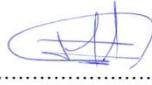
Quito, marzo del 2017

CESIÓN DE DERECHOS DE AUTOR

Nosotros, ANDREA ESTEFANIA HURTADO TERAN, con documento de identificación N° 1723412985, y JONNY JAVIER LLUMIGUSÍN CHAMBA, con documento de identificación N° 1720968203, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación con el tema: "INTERFAZ GRÁFICA PARA LA VISUALIZACIÓN DE UNA RED DE GRAFOS GEOREFERENCIADOS DE LAS PRINCIPALES CALLES DEL CENTRO DE LA CIUDAD DE QUITO", mismo que ha sido desarrollado para optar por el título de INGENIEROS DE SISTEMAS en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestras condiciones de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



.....
ANDREA ESTEFANÍA
HURTADO TERÁN
CI: 1723412985



.....
JONNY JAVIER
LLUMIGUSÍN CHAMBA
CI: 1720968203

Quito, marzo del 2017

DECLARATORIA DE COAUTORÍA DE LA TUTORA

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Artículo académico, con el tema: "INTERFAZ GRÁFICA PARA LA VISUALIZACIÓN DE UNA RED DE GRAFOS GEOREFERENCIADOS DE LAS PRINCIPALES CALLES DEL CENTRO DE LA CIUDAD DE QUITO", realizado por Andrea Estefanía Hurtado Terán y Jonny Javier Llumigusín Chamba, obteniendo un producto que cumpla con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, marzo del 2017



.....
LINA PATRICIA ZAPATA MOLINA
CI: 0501877278

Dedicatoria

Dedico este trabajo en primer lugar a Dios por haberme dado la vida y las fuerzas día tras día para continuar en los momentos más difíciles y guiarme en una etapa más de mis estudios.

A mis amados padres Santiago y Alexandra, por guiarme por el camino del bien, que con gran esfuerzo me entregaron un hermoso regalo que es el estudio, con su amor, consejos y sabiduría supieron hacer de mí la mujer que soy, son y serán siempre el ejemplo de lucha y perseverancia a seguir.

A mis hermosos hijos Skarleth y Keyler que son el motor de mi vida, el amor y cariño que cada día me demuestran me impulsaron para luchar y cumplir mis objetivos. A mi esposo, José Luis el amor de mi vida, que siempre estuvo a mi lado en las buenas y en las malas apoyándome y dándome fuerzas para cumplir con mi meta.

Andrea Estefanía Hurtado Terán

Dedico este trabajo en primer lugar Dios por permitirme realizar mi sueño, por darme fuerzas en los momentos más difíciles en todas las etapas de mi vida.

A mis amados padres Xavier y Esthela, por ser la luz de mi vida, mis amigos y mis guías, mi ejemplo a seguir, por enseñarme excelentes valores morales, dedicación, trabajo duro, por darme las herramientas necesarias para triunfar en la vida.

A mi esposa, Mireya Lissette mi compañera de vida, mi cómplice, mi amiga, el amor de mi vida, le agradezco por el mejor regalo que me pudo hacer, un niño precioso y muy inteligente, mi amado Benjamín quien a su corta edad se ha convertido en la fuente de inspiración el motor y motivo para mi propia superación.

Jonny Javier Llumigusín Chamba

Agradecimiento

Agradecemos de manera especial a la Ing. Lina Patricia Zapata, MsC., tutora de nuestro trabajo, por habernos motivado y orientarnos con sus conocimientos para poder realizar nuestro trabajo

A la Universidad Politécnica Salesiana que nos abrió sus puertas para empezar y culminar con nuestros estudios, a los docentes que tuvimos la oportunidad de conocer, los cuales formaron parte de nuestra vida universitaria y que supieron compartir sus conocimientos para formar excelentes personas y profesionales.

Andrea Estefanía Hurtado Terán
Jonny Javier Llumigusín Chamba

INTERFAZ GRÁFICA PARA LA VISUALIZACIÓN DE UNA RED DE GRAFOS GEOREFERENCIADOS DE LAS PRINCIPALES CALLES DEL CENTRO DE LA CIUDAD DE QUITO.

Lina Zapata¹, Hurtado Estefania², Llumigusín Jonny³

Resumen

En el presente artículo presenta el desarrollo y construcción de un sistema multiagentes, que permite simular el comportamiento de un vehículo, y de un peatón frente al estado (rojo/amarillo/verde) de un semáforo sobre una red vial. Para ello fue necesario considerar el manejo de tres agentes: semáforo, vehículo y peatón, y establecer la comunicación entre dichos agentes, para coordinar el comportamiento de cada uno en función del estado del semáforo. Adicionalmente la aplicación permite seleccionar un punto de origen y un punto de destino que marcan un tramo de la vía sobre la cual se desplazaría el agente vehículo y el agente peatón.

Abstract

This article presents the development and construction of a multi-agent system, which allows the simulation of the behavior of a vehicle, and of a pedestrian in front of the state (red / yellow / green) of a traffic light on a road network. For this, it was necessary to consider the management of three agents: traffic light, vehicle and pedestrian, and establish the communication between these agents, to coordinate the behavior of each one according to the state of the traffic light. Additionally the application allows to select a point of origin and a point of destination that mark a section of the way on which the vehicle agent and the pedestrian agent would move.

Palabras Clave: Agentes, Grafos, Jade, Sistema multiagentes, FIPA.

Keywords: Agents, Graphs, Jade, Multi-Agent System, FIPA

¹Docente de la carrera de Ingeniería en Sistema, Universidad Politécnica Salesiana, sede Quito.

² Estudiante de la carrera de Ingeniería en Sistemas, Universidad Politécnica Salesiana, sede Quito, Autor para correspondencia: ahurtado@est.ups.edu.ec.

³ Estudiante de la carrera de Ingeniería en Sistemas, Universidad Politécnica Salesiana, sede Quito, Autor para correspondencia: jllumigusin@est.ups.edu.ec.

1. Introducción

En los últimos años la Inteligencia Artificial (IA) [15], ha ido evolucionando y ha surgido un nuevo paradigma computacional llamado Programación Orientada a Agentes (POA) [9], que puede ser visto como una especialización de la programación orientada a objetos (POA). Este nuevo modelo de programación encierra conceptos de las teorías de inteligencia artificial dentro de los sistemas distribuidos.

Las aplicaciones basadas en PAO agrupan una colección de componentes llamados agentes que se caracterizan por la autonomía, la proactividad, la capacidad de comunicación, entre otras. Los agentes tienen la capacidad de moverse de un entorno a otro como se puede ver en la figura 1. Al tener comunicación los agentes, permite que al trabajar sobre un objetivo común forman los Sistema Multiagentes.

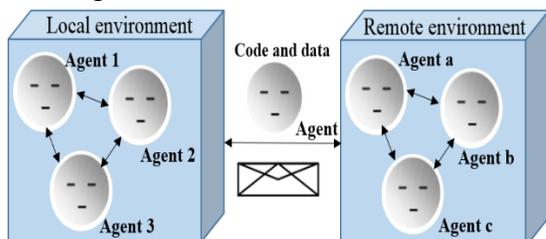


Figura 1. Modelo de un sistema multiagentes [8]

WOOLDRIDGE y JENNINGS (1995) definen a un Agente como un sistema computacional autónomo flexible, que es capaz de actuar en un entorno. “Los agentes constituyen el próximo avance más significativo en el desarrollo de sistemas y pueden ser considerados como la nueva revolución en el software”. Esta frase la dijo Dr. Nicholas Jennings en su discurso al recibir el premio al mejor investigador novel del último congreso internacional de Inteligencia Artificial celebrado en Estocolmo en el año 2000 [3].

Por otra parte, la teoría de grafos, es una herramienta muy útil para el diseño, modelamiento y simulación de datos para diferentes tipos de redes como la red de transporte, redes viales, redes de canalización, encaminamiento de redes de comunicación (Internet, telefonía, etc.), redes eléctricas, entre otras.

Huertas [1] presenta su trabajo en el cual modela una red de tráfico vehicular con distancias y posiciones (vértices y aristas) de las avenidas del cuadro principal de Querétaro, utilizando algoritmos de aproximación Greedy para la optimización de las rutas en la red de tráfico vehicular.

Loarte, Quizhpe y Paz-Arias [2], utiliza el mapa de la ciudad de Loja y establece marcadores en el mapa de Google Maps que representan los semáforos en las intersecciones de las calles. Basado en los algoritmos Dijkstra y el A-star (A*), crearon un algoritmo que calcule el camino más corto desde un nodo origen a un nodo destino teniendo en cuenta los eventos que se pueden producir como accidentes y aglomeración.

El principal objetivo del presente trabajo fue la construcción de una aplicación propia multiagentes conformada por tres agentes: peatón, semáforo y vehículo. Cada agente tiene un comportamiento autónomo, entre los tres se comunican a fin de coordinar cuando actuar en respuesta al comportamiento del otro agente.

El escenario final alcanzado fue el de representar el comportamiento de estos tres agentes, sobre una ruta marcada, para simular el tráfico vehicular en dicha zona. Se utilizó herramientas como el API de Google Maps, QGIS y OpenStreetMap para la obtención de la red de grafos georeferenciadas (longitud y latitud) de las intersecciones y calles de una determinada zona marcada (punto de origen - punto de destino), sobre la cual el agente iniciará su

recorrido hasta llegar a una intersección, donde el agente semáforo se encuentra ubicado, el estado del semáforo (verde/amarillo/rojo) será comunicado al agente vehículo para que este detenga su recorrido si el estado del semáforo está en rojo caso contrario seguirá con su recorrido hasta llegar a su destino.

Para el caso del Peatón procederá de forma similar que el vehículo pero relacionada al cruce de la calle en función del estado del mismo agente semáforo que utiliza el agente vehículo, con la diferencia de que si está en rojo es cruce para el peatón y si está en verde o amarillo es una señal de no cruce.

1.1 Lenguaje de programación y Herramientas

Quantum GIS es un sistema de información geográfica (SIG) de código abierto y distribución libre, ofrece un entorno con múltiples herramientas para el análisis y tratamiento de la información geográfica [4] y **OpenStreetMap** es a la vez un proyecto de cartografía libre y una comunidad de voluntarios que funciona con una mecánica colaborativa. [5]

Red de Grafos: En general, un grafo es una manera de representar relaciones que existen entre pares de objetos. Así, un grafo es un conjunto de objetos, llamados vértices [1], y relaciones entre objetos que establecen una relación entre pares de vértices, representadas por aristas. 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 relación existente entre las calles, intersecciones y las rutas.

Prácticamente cualquier red puede ser modelada con un grafo como: una red de carreteras que conecte ciudades, una red eléctrica o una red de alcantarillado. [6]

Jade: Existen varias plataformas para la implementación de agentes (ej., JADE, Jess, JavaLog, FIPA OS, Voyeguer ORB), y varios protocolos para realizar la comunicación, entre los más conocidos se encuentran Knowledge Query Manipulation Lenguaje (KQML) y (Agent Communication Language) ACL [10]. La plataforma JADE [7] es una de las más usadas y conocidas, esta herramienta permite realizar la implementación de sistemas multiagentes, además permite la coordinación de múltiples agentes FIPA y proporciona una implementación estándar del lenguaje de comunicación FIPA-ACL [11] Agent Communication Language que fue propuesto por FIPA (The Foundation for Intelligent Physical agents), y es un lenguaje estándar para la comunicación entre agentes, fue el sucesor del KQML.

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.

2. Trabajos Relacionados

Grupo de investigación en Software de la Universidad Pedagógica y Tecnología de Colombia realizó una aplicación que simula el comportamiento de un sistema de transporte público colectivo urbano en ciudades de tamaño mediano, a través de la cuantificación de variables complejas que miden la calidad del servicio prestado. Este trabajo les permitió deducir que los agentes de software pueden ser usados en el análisis microscópico de cualquier sistema complejo. [16]

En la Universidad de los Andes (Camacho, et al., en el año 2008) se planteó la problemática actual de

movilidad en la ciudad de Mérida. Con esto se desarrolló un modelo multiagentes en una conocida plataforma de simulación NetLogo, que simula el tráfico de automóviles en una intersección de la ciudad de Mérida, este modelo se adapta a las características de una intersección real, así como a ciertos patrones de comportamiento de los conductores reales. Representando diferentes tipos de autos asignándole un comportamiento particular a cada conductor, representar los cambios de carril, los cruces indebidos y las infracciones ocurridas comúnmente en la situación real. [17]

En la investigación presentada por: Saeed, et al. , en el 2011. Se desarrolló una aplicación de lógica difusa para el control de semáforos autónomos basados en multiagentes. El sistema utiliza sensores inalámbricos los cuales permiten evitar problemas de congestión, accidentes, altas velocidades e irregularidades presentadas en el tráfico. Este enfoque provee una solución minimizando el tiempo de espera de los vehículos de emergencia utilizando un control difuso bajo situaciones que normalmente ocurren en las vías. [18]

3. Metodología

Para el desarrollo del sistema Multiagentes propuesto fue necesario desarrollar tareas o actividades listadas en la figura 2, que se encuentran agrupadas en las siguientes fases.

Fase de Análisis y obtención de datos:

En esta fase se definieron los roles principales que tienen cada agente y se establecieron las relaciones entre cada agente. Los modelos de interacción definen los protocolos a usar, así como los agentes que inician una iteración y los receptores y el procesamiento de información. Los datos adquiridos para la formación de la

red vial, sobre la cual los agentes podrán interactuar, corresponden a las coordenadas georeferenciadas de las intersecciones existentes en las principales calles del sector centro de la ciudad de Quito y dicha información se almacenada en una base de datos.

Fase de diseño y construcción: Es el diseño concreto donde existen los modelos de agentes y el modelo de comunicación representados por el diagrama de secuencia de la figura 6. A través de la librería Jade se desarrolló el sistema multiagentes con su respectiva interfaz gráfica para visualizar el comportamiento de los agentes sobre el mapa de la ciudad de Quito que fue implementado a través del API de Google Maps.

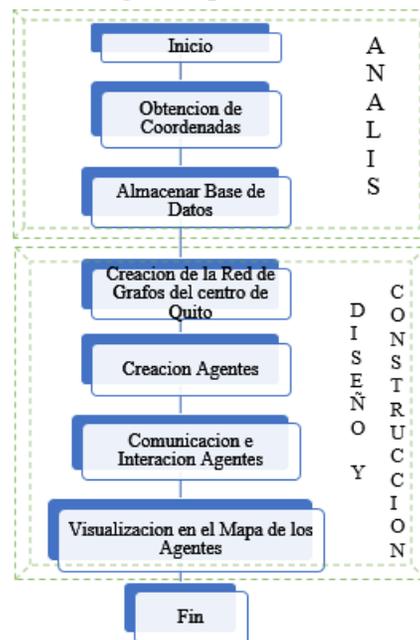


Figura 2. Metodología del desarrollo de la aplicación

4. Diseño y Construcción

4.1 Red de grafos

Para la creación de red de grafos se procedió a cargar el mapa, a través de la herramienta OpenStreetMap, seguidamente se selección el sector centro de la ciudad

de Quito y a través de la generación de una nueva capa shapefile [12] donde se marcaron las coordenadas (latitud y longitud) de los principales puntos que representan las principales calles e intersecciones del centro de la Ciudad, toda esta información se guardó en el archivo con extensión CSV. Todos estos pasos se describen en resumen en la figura 3.

Adicional se creó una base de datos, en Postgresql, donde se almacena información referente al nombre de las calles principales y secundarias, nombres de avenidas junto a sus respectivas coordenadas, nodos y aristas.

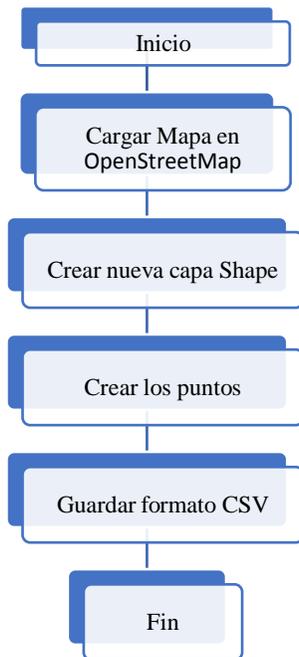


Figura 3. Diagrama muestra la creación del archivo CSV con puntos georeferenciados.

4.2 Visualizador de la red de grafos y del sistema multiagentes

Para la construcción del visualizador para la red de grafos y del sistema multiagentes, se usó el Api de Google para Java que permite heredar algunas de las funciones

del Google Maps como: size, zoom, markets, icons, static y map [13].

Para que exista una adecuada comunicación entre los agentes se definieron tiempos y estados que deben cumplir a fin de dar inicio y finalización a su respectivo comportamiento. En la tabla 1, se pueden observar, en resumen, los diferentes estados y tiempos asignados a cada agente.

Tabla 1. Estados e interacción entre agentes.

Semáforo	Vehículo	Peatón	Tiempo
Rojo	Detener	Avanza	5s
Amarillo	Avanzar	Detener	5s
Verde	Avanzar	Detener	5s

Con la herramienta Eclipse se desarrolló la interface gráfica de la aplicación, haciendo uso de algunos componentes de la herramienta como son:

Image [14].- Despliega la parte gráfica.

Combo Box [14]. - Permite seleccionar la ruta que los agentes se desplazaron.

Button [14].- Encargado de dar inicio a la ejecución del simulador.

4.3 Comportamiento de los agentes

Los tres agentes: semáforo, peatón y vehículo, considerados dentro sistema multiagentes construido, se comunicaron e interactuaron entre sí y cada uno de los agentes tiene un comportamiento diferente.

Donde el agente semáforo, cada determinado tiempo (5s.), cambia su estado de color rojo ha amarillo y a verde o viceversa, el comportamiento del agente peatón dependerá del estado del agente semáforo. Si está en verde o amarillo el agente peatón no se moverá y si está en rojo el agente peatón se moverá, simulando cruzar la calle, de igual forma el comportamiento del agente vehículo dependerá del estado que tenga el agente

semáforo, si está en verde o amarillo el agente vehículo simulará moverse sobre la vía, pero si está en rojo el agente vehículo se detendrá.

Los agentes lograron interactuar y formar un flujo de tráfico por medio de la comunicación que mantienen entre sí, para tomar una decisión el agente vehículo pregunta por su estado al agente semáforo y éste le responde enviándole su estado actual, con estos datos el agente vehículo dará paso a su comportamiento ya definido.

De forma similar, el proceso de comunicación es llevado a cabo entre los agentes peatón y semáforo, a fin de definir el comportamiento del agente peatón. En la figura 4, se describe la forma de comunicación que se lleva a cabo entre los tres agentes a través de un diagrama de secuencia.

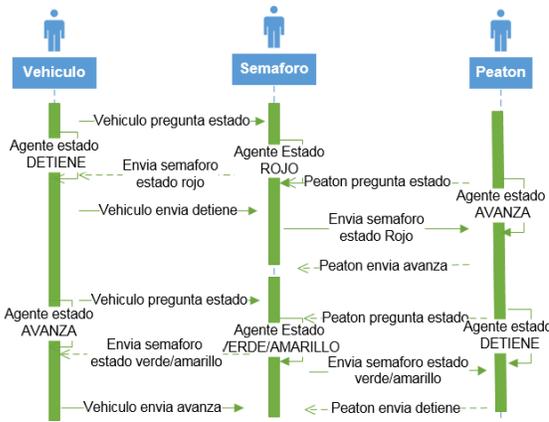


Figura 4. Diagrama secuencia muestra comunicación de agentes

4.4 Comunicación entre agentes

Otra librería que se integró al desarrollo del sistema multiagentes es el Api de JADE, que se la puede considerar como la más importante, ya que permitió la creación y comunicación de los agentes: vehículo, semáforo y peatón. Para dar uso de esta librería se extendió una clase normal y a través de la clase ACLMessage, propia del Api de JADE, que junto a la

clase MessageTemplate se puede establecer el mensaje a enviarse entre los agentes el mismo que está conformado por los siguientes componentes:

```
Datos del remitente.- cfp.addReceiver(s);
Texto del mensaje a ser enviado
cfp.setContent("c");
Envió en si del mensaje
myAgent.send(cfp);
```

La codificación requerida para cada uno de estos componentes tanto del lado del emisor como receptor, siendo el emisor el agente Vehículo y el receptor es el agente semáforo, se detalla a continuación, teniendo en cuenta que son sentencias propias de la librería JADE.

Lo que envía agente vehículo

```
jade.lang.acl.ACLMessage cfp=new
jade.lang.acl.ACLMessage(jade.lang.acl.ACLMessage
.CFP);
for(int
i=0;i<semaforo.length;i++){
    cfp.addReceiver(semaforo[i]);
}
cfp.setContent(colorSemaforo);
cfp.setConversationId("")
cfp.setReplyWith("cfp"+System.currentTimeMillis()
);
myAgent.send(cfp);
mt =
MessageTemplate.and(MessageTemplate.MatchConversa
tionId(""),
MessageTemplate.MatchInReplyTo(cfp.getReply
With()));
step=1;
```

Lo que el agente semáforo responde.

```
MessageTemplate mt =
MessageTemplate.MatchPerformative(jade.lang
.acl.ACLMessage.CFP);
jade.lang.acl.ACLMessage
msg=myAgent.receive(mt);
if(msg !=null){
    String coordenada = msg.getContent();
    jade.lang.acl.ACLMessage respuesta =
msg.createReply();
    Integer estado = (Integer)
catalogo.get(coordenada);
    if(estado != null){
        respuesta.setPerformative(jade.lang.acl.ACL
Message.PROPOSE);
        respuesta.setContent(String.valueOf(String.
valueOf(estado.intValue())));
    }else{
```

```

respuesta.setPerformative(jade.lang.acl.ACL
Message.REFUSE);
respuesta.setContent("No
disponible");
}

```

El agente vehículo recibió el mensaje de respuesta y lo interpretó.

```

jade.lang.acl.ACLMessage
reply=myAgent.receive(mt);
if(reply != null){
if(reply.getPerformative() ==
jade.lang.acl.ACLMessage.PROPOSE){
int colorSemaforo =
Integer.parseInt(reply.getContent());

if(colorSemaforo == 1){
EstadoSemaforo = "green";
}else if(colorSemaforo == 2){
EstadoSemaforo = "yellow";
}else if(colorSemaforo == 3){
EstadoSemaforo="red";
}
}
}

```

Para iniciar el comportamiento de cada agente la librería JADE cuenta con un gestor de agentes ver figura 5, este administra el comportamiento de cada agente como: el de iniciar, detener y muerte del agente.

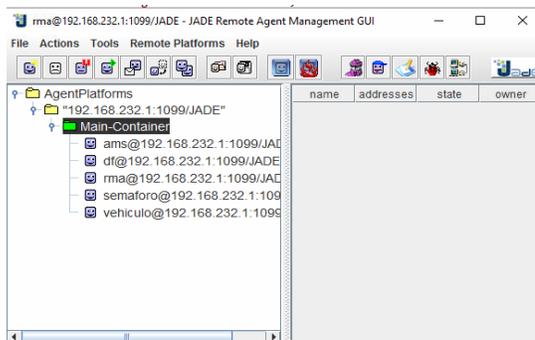


Figura 5. Interface para administrar agentes

5. Pruebas y Resultados

5.1 Generación de la red de grafos:

La obtención de la red de grafos georeferenciada, sobre las intersecciones de las principales calles del centro de la ciudad, se logró a través de la obtención de la longitud y latitud de cada uno de los puntos que conforma dicha red, como se puede observar en la figura 6.



Figura 6. Red de Grafos sobre las calles del centro de la ciudad de Quito

5.2 Comunicación entre agentes (vehículo, semáforo y peatón)

Los agentes vehículo, semáforo y peatón se mantienen en comunicación permanente mediante el framework JADE. El agente semáforo presenta el estado verde o amarillo envía los datos a los agentes peatón y vehículo ya que estos dependen de su comportamiento para tomar una decisión. El agente peatón recibe los datos del agente semáforo y su comportamiento fue el detenerse mientras que el agente vehículo al recibir dichos datos procede avanzar. Como se puede ver en la figura. 7 y figura 8 respectivamente.

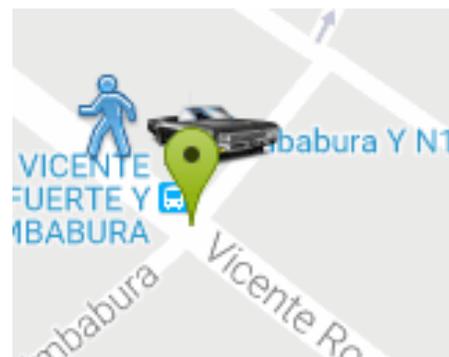


Figura 7. Prueba Agente semáforo estado Verde

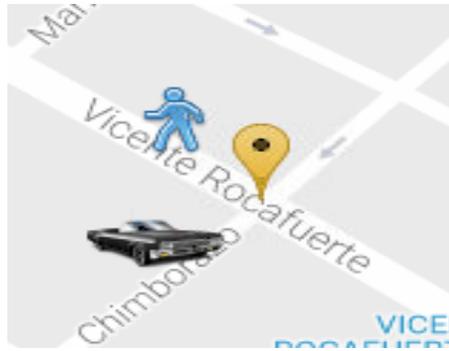


Figura 8. Prueba Agente semáforo estado Amarillo

Mientras que, si el agente semáforo cambia su estado a Rojo, el agente peatón procede avanzar y el agente vehículo se detiene, como se puede ver en la figura 9, dichos agentes actúan en forma coordinada y eso se debe a la buena comunicación que manejan entre sí.

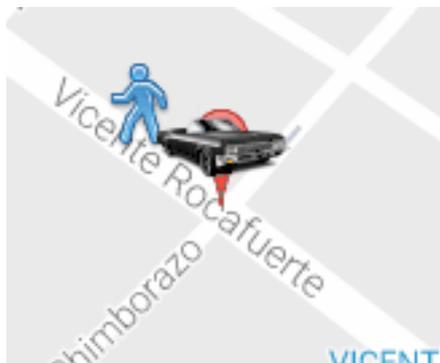


Figura 9. Prueba Agente semáforo estado Rojo

En la tabla 2 se puede observar los pasos a considerar para verificar el consumo de recursos al ejecutar la aplicación.

Tabla 2. Pasos a seguir para verificar el uso de recursos por la aplicación.

Pasos	Descripción
1	Antes de la apertura de la aplicación
2	Iniciar agentes a utilizarse
3	Ejecutar la aplicación, desplazamiento y comunicación de agentes

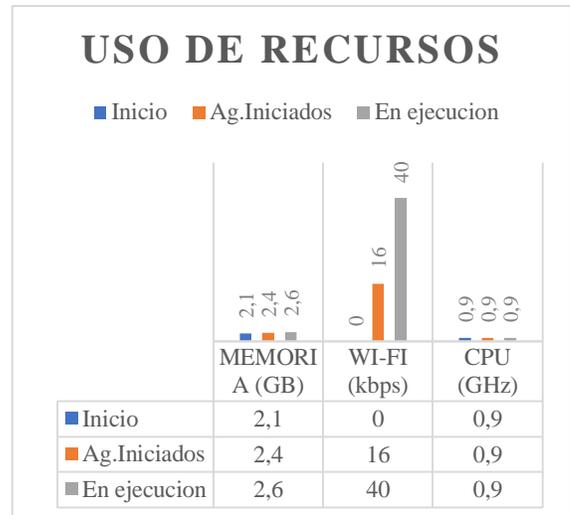


Figura 10. Muestra el consumo de recursos

La figura 10, describe la variación del consumo de recursos durante la ejecución de la aplicación. Donde se puede apreciar que el consumo de memoria aumenta 2.1 GB a 2.6 al ejecutar por completo la aplicación. Mientras que el WI-FI tiene un gran cambio de 0kbps a 40kbps, esto es debido a las peticiones que genera la aplicación al usar el Api de Google Maps.

6. Conclusiones

Los agentes lograron desplazarse por el mapa georeferenciado del centro de la ciudad de Quito sin ningún problema, esto fue posible ya que la red de grafos se levantó correctamente. Es decir, no se perdieron puntos ni se incrementó puntos. Lo que demuestra que QGIS es una herramienta confiable para la creación de una red Grafos.

La red de grafos georeferenciada obtenida a través de la librería geotools requirió que las posiciones (longitud, latitud) de las intersecciones de las principales calles del centro de la ciudad, sean obtenidas una a una por medio de la herramienta QGIS. Lo que demuestra que las herramientas se complementan.

La creación de sistemas multiagentes demanda mucho recurso del ordenador debido a que cada agente se encuentra en ejecución y más aún si existe comunicación entre ellos ya que demanda que permanentemente estén en vigilia a la llegada de mensajes o peticiones que enseguida son respondidas.

La comunicación establecida entre agentes se logró de manera exitosa ya que en el visualizador se puede notar claramente que el agente vehículo asume su comportamiento de acuerdo al estado que reporta el agente semáforo y lo mismo ocurre con el agente peatón.

7. Trabajos Futuros

Para posteriores trabajos, se debería optar por agregar más agentes que interactúen en distintas calles y se comuniquen entre todos. Por otro lado, se podría considerar el manejo de rutas que recorran distintas calles tomando en cuenta el sentido de las vías, ya sea una sola vía o doble vía y evitar el desplazamiento en contra vía.

Referencias

- [1] Huerta Jiménez, H. M. (2014). *Optimización de rutas en redes de tráfico vehicular* (Doctoral dissertation).
- [2] Caraguay, L., Iván, R., & Vásquez, B. R. Q. (2016). Desarrollo y simulación de un sistema multiagente para la comunicación de semáforos para encontrar la ruta óptima mediante grafos (Bachelor's thesis).
- [3] Jennings, N. R., & Wooldridge, M. (1998). Applications of intelligent agents. In *Agent technology* (pp. 3-28). Springer Berlin Heidelberg.
- [4] Morales Ramírez, J. M. (2013). Sistemas de Información Geográfica: Servidor de Mapas de la Comunidad, El Conejo, Municipio de Perote, Veracruz, utilizando software libre.
- [5] Ortega, I. S., & Jové, J. F. (2008). Software geoespacial por y para OpenStreetMap. *way*, 501, 23151177.
- [6] Tecnológico de Monterrey, Teoría de Grafos; Recuperado el 04 de septiembre del 2016 a partir de http://campus.cva.itesm.mx/nazira/Tc1003/PDF/TODO/0701_Tc1003_TODO_Grafos.pdf
- [7] JADE: Java Agent Development Framework. <http://jade.tilab.com.3/2007.t>
- [8] Avancini, H., & Amandi, A. SADIO Electronic Journal of Informatics and Operations Research.
- [9] Diego Caviedes,(2010), Agentes Inteligetes, Programacion Orientada a Agentes; Recuperado el 10 de enero del 2017 desde: <http://agentesjade.blogspot.com/2010/01/programacion-orientada-agentes.html>
- [10] Programación JADE, Recuperado el 10 de enero del 2017 desde: <https://programacionjade.wikispaces.com/Comunicaci%C3%B3n>
- [11] JADE: Java Agent Development Framework. http://jade.tilab.com/papers/JADETutorialIEEE/JADETutorial_FIPA.pdf
- [12] ArcGIS: <http://desktop.arcgis.com/es/arcmap/10.3/manage-data/shapefiles/what-is-a-shapefile.htm>
- [13] API DE GOOGLE MAPS: <https://developers.google.com/maps/documentation/javascript>
- [14] ECLIPSE: <http://help.eclipse.org/kepler/index.jsp?topic=%2FForg.eclipse.jubula.client.ua.help%2Fhtml%2Freference%2Fnode14.html>
- [15] Luigi Ceccaroni, (Primavera 2007) Inteligencia Artificial,

Introducción a la inteligencia artificial; Recuperado el 08 de enero del 2017 desde: [http://www.cs.upc.edu/~luigi/II/IA-2007-fall/1-introduccion-a-la-inteligencia-artificial-\(es\).pdf](http://www.cs.upc.edu/~luigi/II/IA-2007-fall/1-introduccion-a-la-inteligencia-artificial-(es).pdf)

- [16] Callejas-Cuervo, M., Valero-Bustos, H., & Alarcón-Aldana, A. (2014). Agentes de software como herramienta para medir la calidad de servicio prestado en un sistema de transporte público colectivo urbano. *Información tecnológica*, 25(5), 147-154.
- [17] Camacho, J., Medina, S., & Terán, O. (2012). Simulación del tráfico de autos en una intersección: desde la perspectiva de una plataforma multiagente. *Ciencia e Ingeniería*, 33(2), 85-94.
- [18] Saeed, Y., Khan, M. S., Ahmed, K., & Mubashar, A. S. (2011). A multi-agent based autonomous traffic lights control system using fuzzy control. *International Journal of Scientific & Engineering Research*, 2(6), 1.

