

**MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL
PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA**



JULIÁN ALBERTO ESPEJO DÍAZ

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

FACULTAD DE INGENIERÍA

PROYECTO CURRICULAR INGENIERÍA INDUSTRIAL

GRUPO DE INVESTIGACIÓN SISTEMAS EXERTOS Y SIMULACIÓN (SES)

BOGOTÁ D.C.

2016

**MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL
PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA**

Trabajo de grado presentado como requisito para obtener el título de:

INGENIERO INDUSTRIAL

Modalidad investigación

Presentado por:

JULIÁN ALBERTO ESPEJO DÍAZ

CÓDIGO: 20111015046

Dirigida por:

PH.D. GERMÁN ANDRÉS MÉNDEZ GIRALDO

Co-dirigida por:

M.SC. EDUYN RAMIRO LÓPEZ SANTANA

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

FACULTAD DE INGENIERÍA

PROYECTO CURRICULAR INGENIERÍA INDUSTRIAL

GRUPO DE INVESTIGACIÓN SISTEMAS EXERTOS Y SIMULACIÓN (SES)

BOGOTÁ D.C.

2016

Resumen

El cuidado a la salud domiciliaria se trata básicamente de una forma de prestación de servicio médico en el cual personal capacitado atiende las necesidades en materia de salud de los pacientes en sus hogares. La prestación de éste servicio implica realizar el ruteo de personal, es decir, determinar la secuencia que los operadores de salud deben seguir para visitar a los pacientes que le fueron asignados. Lo anterior debe ser realizado teniendo en cuenta múltiples criterios de optimización propios del cuidado a la salud domiciliaria. Por otra parte también se debe tener en cuenta que el cuidado domiciliar es de naturaleza dinámica debido a que nuevos pacientes pueden requerir del servicio aun cuando los operadores de salud ya estén visitando otros pacientes. Para responder a lo anterior se debe realizar varios esquemas de ruteo en el mismo horizonte de planeación con el fin de que se atienda lo mejor posible a los nuevos pacientes. Este trabajo presenta un modelo de simulación basado en agentes que permite realizar el ruteo de operadores de salud aceptando nuevas demandas durante un horizonte de planeación. El ruteo es resuelto mediante un modelo matemático propuesto que busca minimizar los tiempos de transporte de los operadores de salud así como la promesa de servicio a los pacientes. El modelo de simulación fue implementado utilizando la librería Jade de Java, se realizaron varias pruebas para conocer el desempeño del modelo de simulación y finalmente realiza un estudio de caso de un proveedor del servicio de cuidado domiciliar en la ciudad de Bogotá.

Palabras clave: ruteo de personal, cuidado a la salud domiciliaria, modelamiento matemático, sistemas multi-agente, simulación, Jade.

Abstract

Home Health care is a way of providing health services in which the medical staff commutes to the patients' homes in order to meet their medical needs. Providing home health care services implies the routing of caregivers, which means to decide in which sequence each caregiver will visit the patients assigned to him/her. The routing of caregivers have to be done considering several optimization criteria and also considering the dynamism of home health care services. The dynamism of this service consist in the fact that new request may appear while the system is already running, therefore multiple routing scheme have to be done in the same planning horizon to better serve the new requests. This document presents an agent based simulation model which solves dynamically the caregivers routing problem accepting new request during a planning horizon. The routing schemes are made using a proposed mathematical model which aims to minimize the travel time of the caregivers and also aims to minimize the service promise to the patients. The agent based simulation model were built using a Java library named Jade, the model were tested several times in order to know the performance of the model. Finally a case study of a real provider of home health care services in Bogota is presented in order to validate the previous work.

Key words: Routing staff, home Health care services, mathematical modeling, multi-agent system, agent based simulation, Jade.

CONTENIDO

1. INTRODUCCIÓN	1
2. PLANTEAMIENTO DEL PROBLEMA DE INVESTIGACIÓN	3
2.1. DESCRIPCIÓN DEL PROBLEMA.....	3
2.2. OBJETIVOS.....	5
2.2.1. Objetivo general.....	5
2.2.2. Objetivos específicos.....	5
2.3. ALCANCE.....	5
2.4. JUSTIFICACIÓN.....	6
2.5. METODOLOGÍA.....	7
3. REVISIÓN DEL ESTADO DEL ARTE	9
3.1. REVISIÓN DE LA LITERATURA EN ASIGNACIÓN Y RUTEO DE OPERADORES DE SALUD EN EL CUIDADO DE LA SALUD DOMICILIARIA	9
3.1.1. Metodología para la revisión del estado del arte	10
3.1.2. Problemas de asignación vs problemas de asignación y ruteo.....	10
3.1.3. Problemas estocásticos vs problemas determinísticos.....	11
3.1.4. Funciones objetivo y/o medidas de desempeño	11
3.1.5. Restricciones del problema	14
3.1.6. Métodos de solución.....	17
3.1.7. Datos de prueba	19
3.1.8. Consideraciones de planeación dinámica	20
3.1.9. Software específico para resolver problemas de cuidado domiciliario	21
3.2. CONCEPTUALIZACIÓN DE SIMULACIÓN BASADA EN AGENTES	22
3.2.1. Metodologías para la construcción de modelos de simulación de sistemas basados en agentes.....	23
3.2.2. Plataformas para el desarrollo de sistemas multi-agente	26
3.2.3. Revisión de aplicaciones de sistemas multi-agente.....	27
3.3. CONCLUSIONES DEL CAPÍTULO	28
3.4. RESULTADOS DE INVESTIGACIÓN	30
4. MODELO MATEMÁTICO PARA EL RUTEO DE OPERADORES DE SALUD EN EL CUIDADO DOMICILIARIO	31
4.1. DEFINICIÓN DEL PROBLEMA	31

*MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL
PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA*

4.2.	DESARROLLO DEL MODELO MATEMÁTICO.....	33
4.2.1.	Notación del problema.....	33
4.2.2.	Formulación del problema.....	34
4.3.	IMPLEMENTACIÓN DEL MODELO MATEMÁTICO.....	36
4.4.	EXPERIMENTACIÓN Y RESULTADOS.....	38
4.5.	CONCLUSIONES.....	42
4.6.	RESULTADOS DE INVESTIGACIÓN.....	42
5.	APROXIMACIÓN MULTI-AGENTE DEL PROBLEMA DE RUTEO EN EL CUIDADO DOMICILIARIO	43
5.1.	ETAPA DE CONCEPTUALIZACIÓN DEL SISTEMA.....	43
5.1.1.	Definición de las entidades.....	44
5.1.2.	Definición de las interacciones.....	45
5.2.	ETAPA DE ANÁLISIS DEL SISTEMA.....	47
5.2.1.	Actividades.....	47
5.2.2.	Comunicación e interacción.....	48
5.3.	ETAPA DE DISEÑO DEL SISTEMA.....	49
5.3.1.	Modelo de agentes.....	49
5.3.2.	Reglas del sistema basado en agentes.....	50
5.4.	CONCLUSIONES DEL CAPÍTULO.....	51
6.	IMPLEMENTACIÓN DEL SISTEMA MULTI-AGENTE DE CUIDADO DOMICILIARIO.....	52
6.1.	SOFTWARE EMPLEADO.....	52
6.1.1.	Java y Eclipse.....	52
6.1.2.	Librerías.....	52
6.2.	DIAGRAMAS DE CLASE DEL MODELO BASADO EN AGENTES.....	54
6.2.1.	Clase AgenteGenerador.....	54
6.2.2.	Clase AgenteCoordinador.....	55
6.2.3.	Clase LeerDatos.....	56
6.2.4.	Clase Modelomatemat.....	57
6.2.5.	Clase AgenteOperador.....	58
6.3.	SECUENCIA DE LA EJECUCIÓN DEL CÓDIGO COMPUTACIONAL.....	59
6.4.	PRUEBAS DEL CÓDIGO COMPUTACIONAL.....	63
6.4.1.	Prueba 1 – Experimentación y resultados.....	63
6.4.2.	Prueba 2 – Experimentación y resultados.....	65
6.4.3.	Prueba 3 – Experimentación y resultados.....	66
6.4.4.	Prueba 4– Experimentación y resultados.....	68
6.5.	CONCLUSIONES DEL CAPÍTULO.....	69

*MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL
PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA*

6.6.	RESULTADOS DE INVESTIGACIÓN	70
7.	ESTUDIO DE CASO	71
7.1.	DESCRIPCIÓN DE LA SITUACIÓN.....	71
7.2.	INFORMACIÓN Y ANÁLISIS DE ENTRADA DEL SISTEMA	73
7.2.1.	Análisis de entrada de las solicitudes por hora del sistema.....	73
7.2.2.	Frecuencia de solicitudes por localidad	74
7.2.3.	Georreferenciación de los pacientes.....	74
7.3.	CONFIGURACIÓN DE LA SIMULACIÓN BASADA EN AGENTES	75
7.3.1.	Calculo de las distancias y tiempos de desplazamiento entre nodos	75
7.3.2.	Parámetros del modelo de simulación.....	76
7.4.	RESULTADOS DEL ESTUDIO DE CASO	78
7.5.	CONCLUSIONES DEL ESTUDIO DE CASO	80
	CONCLUSIONES, TRABAJO FUTURO Y VALIDACIÓN DE LOS OBJETIVOS DE INVESTIGACIÓN.	81
	REFERENCIAS.....	84
	ANEXOS	91
	Anexo 1: clasificación de la revisión bibliográfica consolidada.....	91
	Anexo 2: Código de Xpress del modelo matemático	93
	Anexo 3: Código construido en java del modelo de simulación basado en Agentes.....	96
	Anexo 4: información histórica de solicitudes por horas.....	115
	Anexo 5: Análisis de entrada –Bondad de ajuste de las solicitudes por horas	118
	Anexo 6: Asignación y ruteo en el cuidado de la salud domiciliaria: revisión de la literatura	
	Anexo 7:Modelo de programación entera mixta para programación y ruteo en el cuidado a la salud domiciliaria considerando la promesa de servicio	
	Anexo 8: Multi-agent approach for solving the dynamic home health care routing problem	

Lista de Figuras

Figura 1 Metodología del proyecto de investigación.....	7
Figura 2 Diagrama de flujo de la estructura metodológica de la investigación.....	8
Figura 3 Metodología Simulación basada en Agentes de Macal & North	24
Figura 4 Metodología de simulación de sistemas basados en agente de Moreno.....	25
Figura 5 Función cuadrática del TRIAGE.....	32
Figura 6 Minimización de la promesa de servicio	32
Figura 7 Problema de ruteo de cuidado domiciliario.....	33
Figura 8 Diagrama de casos de uso del problema de cuidado domiciliario.....	46
Figura 9 Diagrama de actividades del sistema de cuidado domiciliario	47
Figura 10 Diagrama de secuencia del sistema de cuidado domiciliario.....	48
Figura 11 Diagrama de árbol de agentes	49
Figura 12 Configuración de la aproximación multi-agente	50
Figura 13 Configuración de Jade en Eclipse	53
Figura 14 Diagrama de clase para la clase AgenteGenerador.....	54
Figura 15 Diagrama de clase para la clase AgenteCoordinador.....	55
Figura 16 Diagrama de clase para la clase LeerDatos	56
Figura 17 Diagrama de clase para la clase Modelomatemat	57
Figura 18 Diagrama de clase para la clase AgenteOperador	58
Figura 19 Inicialización de los Agentes.....	59
Figura 20 Datos para el primer periodo en la prueba del código	59
Figura 21 Solución del problema de ruteo en Java	60
Figura 22 Pacientes a visitar, tiempos de llegada y disponibilidad de cada operador	61
Figura 23 Notificación del tiempo y momento de disponibilidad.....	61
Figura 24 Generación de datos del periodo 2 en adelante.....	62
Figura 25 Configuración de la prueba de ejecución 1	64
Figura 26 Configuración de la prueba de ejecución 3	66
Figura 27 Coordenadas de los pacientes por localidades	75
Figura 28 Distancia esférica y distancia real	76

Lista de Tablas

Tabla 1 Clasificación de referencias revisadas según componente abordado	10
Tabla 2 Clasificación de referencias revisadas según consideración de la incertidumbre	11
Tabla 3 Clasificación de referencias revisadas según la función objetivo.....	14
Tabla 4 Clasificación de las referencias revisadas según sus restricciones.....	16
Tabla 5 Clasificación de las referencias revisadas según el método exacto utilizado.....	18
Tabla 6 Clasificación de referencias según el método aproximada utilizado	18
Tabla 7 Clasificación de las referencias revisadas según los datos de prueba.....	20
Tabla 8 Tiempo de atención con base en el nivel de TRIAGE.....	32
Tabla 9 Notación usada para el modelo de programación entera mixta.....	34
Tabla 10 Información de operadores de salud (pr01).....	38
Tabla 11 Información de pacientes (pr01)	38
Tabla 12 Resultados de la prueba variando α y β para instancia pr01	39
Tabla 13 Esquema de ruteo para $\alpha=0.2$ $\beta=0.8$ instancia pr01.....	39
Tabla 14 Esquema de ruteo para $\alpha=0.8$ $\beta=0.2$ instancia pr01	39
Tabla 15 Información de operadores de salud instancia pr02.....	40

*MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL
PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA*

Tabla 16 Información de pacientes instancia pr02	40
Tabla 17 Resultados de la prueba variando α y β para instancia pr02	40
Tabla 18 Esquema de ruteo para $\alpha=0.2$ $\beta=0.8$ instancia pr02.....	41
Tabla 19 Esquema de ruteo para $\alpha=0.8$ $\beta=0.2$ instancia pr02.....	41
Tabla 20 Plantilla de caso de uso del problema de cuidado domiciliario	46
Tabla 21 Parámetros de la prueba de ejecución 1.....	64
Tabla 22 Resultados de la prueba de ejecución 1.....	65
Tabla 23 Resultados de la prueba de ejecución 2.....	65
Tabla 24 Parámetros de la prueba de ejecución 2.....	67
Tabla 25 Resultados de la prueba de ejecución 3.....	67
Tabla 26 Resultados de la prueba de ejecución 4.....	68
Tabla 27 Usuarios del servicio de cuidado domiciliario por localidad	71
Tabla 28 Distribuciones de solicitudes por horas.....	73
Tabla 29 Frecuencia de solicitudes por localidad.....	74
Tabla 30 Parámetros del estudio de caso	76
Tabla 31 Información de entrada de los operadores de salud en el estudio de caso	77
Tabla 32 Información de los paciente del estudio de caso	77
Tabla 33 Resultados estudio de caso	78
Tabla 34 Rutas para los operadores de salud estudio de caso $\alpha=0.8$ $\beta=0.2$	79
Tabla 35 Validación de los objetivos del proyecto de investigación.....	82
Tabla 36 Nomenclatura de la consolidación de la revisión bibliográfica	91
Tabla 37 Clasificación consolidada de las referencias revisadas.....	92
Tabla 38 Información histórica de las solicitudes por horas.....	115

Capítulo 1

INTRODUCCIÓN

El cuidado a la salud domiciliaria es una forma de prestación de servicio médico en el cual personal capacitado atiende las necesidades en materia de salud de los pacientes en sus hogares. En la literatura es conocida como *Home Health Care* y se trata de cualquier cuidado o atención ya sea médica o social que permita a las personas permanecer en sus viviendas [1], [2]. El sector del cuidado domiciliario está experimentando un rápido crecimiento debido al constante aumento y envejecimiento de la población así como al incremento de la expectativa de vida de las personas [3], [4]. Lo anterior resulta en un aumento de la demanda de servicios de salud y específicamente de los servicios de cuidado domiciliario [5], [6]. Por otra parte éste sector enfrenta presiones gubernamentales para recortar presupuesto y/o problemas financieros, por lo que debe buscar alternativas que le permitan reducir costos en la prestación del servicio utilizando eficientemente los recursos con que cuenta [7], [8]. Para hacer frente a las condiciones anteriormente descritas se deben desarrollar modelos y/o métodos que le permitan optimizar sus operaciones así como desarrollar herramientas efectivas que apoyen sus procesos de decisión. Lo anterior es realizado con el fin de satisfacer las necesidades de los pacientes de una manera efectiva a un bajo costo [9]. Una de las decisiones críticas que se debe tomar frecuentemente en el contexto del cuidado domiciliario es el ruteo de operadores de salud, es decir, determinar la secuencia que un operador de salud debe seguir para visitar a los pacientes asignados a él o a ella [10].

Por otra parte el problema de cuidado a la salud domiciliario puede ser clasificado como estático o dinámico. En su versión estática toda la información de entrada del problema es dada por adelantado, es decir, se conoce el número total de pacientes a atender, su ubicación y sus condiciones antes de resolver el problema por lo que éste debe ser resuelto una sola vez para un horizonte de planeación. Por el contrario, en la versión dinámica del problema existe información que depende del tiempo, es decir, nuevos pacientes pueden surgir en un horizonte de planeación, los pacientes pueden cambiar sus ubicaciones geográficas o pueden requerir una atención más especializada, entre otros. Lo anterior ocasiona que se deban generar varios esquemas de ruteo dentro del mismo horizonte de planeación, los cuales tengan en cuenta los nuevos pacientes o los cambios solicitados por parte de los pacientes. La versión dinámica del problema es la que más se ajusta a la realidad, pues un paciente que solicite el servicio de cuidado domiciliario busca que se le atienda lo más pronto posible, que se le tenga en cuenta su prioridad en la atención y además prefiere que no tenga que esperar hasta el siguiente horizonte de planeación para ser incluido en las atenciones médicas debido a que su solicitud puede ser urgente.

El propósito de éste trabajo es realizar un modelo de simulación basado en agentes que permita resolver de manera dinámica el problema de ruteo de operadores de salud, en donde, mediante la aproximación multi-agente se permita aceptar nuevas demandas en el sistema cuando esté en ejecución. Por otra parte se pretende mediante el modelo matemático utilizado en la aproximación multi-agente minimizar el tiempo de transporte de los operadores de salud así como la minimización de la promesa de servicio a los pacientes, es decir, minimizar el tiempo en que los operadores de salud llegan a realizar la visita domiciliaria teniendo en cuenta la prioridad en la atención de los pacientes. De esa manera se logra lo siguiente:

- ✓ Aumentar el nivel de servicio del proveedor de cuidado domiciliario.
- ✓ Aumentar la satisfacción del paciente debido a que se busca atenderlos lo más antes posible con base en su prioridad.
- ✓ Reducir costos en la prestación del servicio mediante la mejor utilización de los recursos con que cuenta el proveedor.

El documento se organiza de la siguiente manera. El **capítulo 2** presenta el planteamiento del problema de investigación en donde se encuentra la descripción, los objetivos, el alcance del trabajo realizado, la justificación y la metodología utilizada. En el **capítulo 3** se encuentra la revisión del estado del arte en asignación y ruteo de operadores de salud, para lo cual se utilizó una metodología de revisión de literatura sistemática en donde se analizaron los principales objetivos, restricciones, métodos de solución y otras consideraciones identificando las tendencias y los vacíos pertinentes sobre los cuales se desarrolló la investigación. Por otra parte en éste capítulo también se encuentra una breve conceptualización sobre simulación basada en agentes con el fin de que el lector se familiarice con el tema. En el **capítulo 4** se encuentra el modelo matemático del problema en el cual están las funciones objetivo, las restricciones, algunas consideraciones y supuestos realizados, la implementación en un software específico, la experimentación y finalmente el análisis de los resultados obtenidos. En el **capítulo 5** se presenta el diseño, la conceptualización y el análisis de la aproximación multi-agente del problema de cuidado domiciliario. Para ello se utilizan diagramas de caso de uso, de actividades, de secuencia, entre otros, con el fin de establecer la interacción y la comunicación de los agentes dentro del sistema de cuidado a la salud domiciliaria. El **capítulo 6** muestra la implementación de la aproximación multi-agente en un software específico (Java) que fue hecha mediante la construcción y desarrollo del código computacional, finalmente en éste capítulo se realizan pruebas del código con el fin de mostrar la ejecución, el desempeño y el alcance de lo construido en Java. En el **capítulo 7** se realiza el estudio de caso de un proveedor del servicio de cuidado domiciliario en Bogotá, para lo cual se cuenta con información histórica del sistema además de la ubicación real de los usuarios en la ciudad. El estudio de caso se realiza con el fin de demostrar que se puede ingresar información real al modelo de simulación basado en agentes. Finalmente se presentan las conclusiones, los lineamientos sobre trabajos futuros en el tema y la validación de los objetivos del proyecto de investigación.

Capítulo 2

PLANTEAMIENTO DEL PROBLEMA DE INVESTIGACIÓN

2.1. DESCRIPCIÓN DEL PROBLEMA

El ruteo de operadores de salud es una actividad compleja realizada en su mayoría de forma manual en la cual múltiples criterios de optimización tienen que ser tenidos en cuenta. El propósito de esta investigación es realizar la asignación y el ruteo de operadores de salud de manera dinámica, reduciendo los tiempos de desplazamiento de los operadores de salud así como el tiempo de llegada a los hogares de los pacientes con base en la prioridad en su atención. Los objetivos anteriores deben ser cumplidos teniendo en cuenta varias restricciones, como el tiempo de trabajo de los operadores de salud, el envío de operadores de salud calificados a realizar las visitas, la salida y el regreso de los operadores de salud a sus respectivos hogares, entre otras. Es importante resaltar que el componente dinámico del problema yace en la aceptación de nuevos pacientes o nuevas demandas cuando el sistema esté corriendo, en este caso un nuevo esquema de ruteo debe ser generado con el fin de atender a los nuevos pacientes.

En el problema se considera una serie de pacientes que necesitan del servicio de cuidado a la salud domiciliaria. Cada servicio o visita está caracterizada por un nivel de habilidad o experticia determinado y una duración específica. Se considera también la prioridad en la atención con el fin de visitar lo más pronto posible a los pacientes con condiciones médicas más críticas. La prioridad en el servicio toma valores desde 1 (servicio urgente) hasta 5 (servicio menos urgente), por lo cual se busca que los tiempos de llegada a las ubicaciones de los pacientes coincidan con los tiempos establecidos en el TRIAGE disponibles en [11]. Es importante aclarar que el TRIAGE es un concepto utilizado en emergencias médicas el cual establece los tiempos de respuesta de acuerdo a la prioridad de los pacientes. Por consiguiente se pretende enviar operadores de salud calificados a los hogares de los pacientes respetando los tiempos establecidos por el TRIAGE en lo posible mientras se minimizan los tiempos de transporte. Un ejemplo de lo anterior es que se recomienda que a un paciente con prioridad tres se le atienda en un tiempo de 60 minutos, por lo tanto un operador de salud calificado debe llegar aproximadamente en ese tiempo a la ubicación del paciente.

Por otra parte, los operadores de salud empiezan y terminan sus rutas o recorridos en sus hogares, tienen un tiempo máximo de trabajo determinado y están caracterizados por un nivel de habilidad o experticia " q ", el cual toma valores desde 1 (cuidados usuales) hasta 5 (atención especializada). La visita domiciliaria es realizada solo si la habilidad del recurso (operador de salud) es mayor o igual a la habilidad requerida por el paciente, en otras palabras, cada operador de salud con un nivel de habilidad " q " solo puede visitar pacientes que necesitan una habilidad menor o igual a " q ".

Además de lo mencionado anteriormente para efectos del problema se hacen consideraciones y suposiciones que son enunciadas a continuación.

- Si en un horizonte de planeación un paciente requiere dos o más visitas domiciliarias se creará una nueva demanda.
- Los tiempos de viaje son determinísticos.
- Los operadores de salud trabajan en la misma área geográfica y pueden operar en todo el territorio de los pacientes.
- Los operadores de salud inician el servicio inmediatamente cuando llegan a la ubicación de los pacientes.
- Los tiempos de viaje de una ubicación a otra son los mismos sin importar la dirección de viaje.

Como se mencionó anteriormente se considera la posibilidad del ingreso de nuevas demandas cuando el sistema se esté ejecutando, es decir, atender a un nuevo paciente aun cuando eso signifique realizar los esquemas de ruteo de nuevo. Para lograr lo anterior se propone un modelo de simulación basado en agentes el cual mediante la comunicación y la interacción de los agentes que componen el sistema se pueda atender ese nuevo paciente y los existentes en el sistema de la mejor forma.

Es de esa manera que se pretende elaborar un modelo basado en agentes para el apoyo de la toma de decisiones de asignación y ruteo de operadores de salud en el entorno del cuidado de salud en casa, con el fin de darle solución a la necesidad de los proveedores de salud domiciliaria de tomar buenas decisiones a nivel táctico y operativo. Por tanto se plantea la siguiente pregunta de investigación:

¿Cuál debe ser la estructura de un modelo de simulación basado en agentes que realice la asignación y ruteo de operadores de salud teniendo en cuenta las características del entorno de cuidado domiciliario?

2.2. OBJETIVOS

2.2.1. Objetivo general

Proponer un modelo de simulación basado en agentes que permita resolver el problema de asignación y ruteo de operadores de salud en un entorno de cuidado domiciliario.

2.2.2. Objetivos específicos

- ✓ Definir las características teóricas y técnicas del problema de asignación y ruteo en el cuidado de salud en casa.
- ✓ Desarrollar un modelo basado en agentes para la asignación y ruteo de operadores de salud.
- ✓ Implementar el modelo de simulación basada en agentes en un software especializado para ello.
- ✓ Evaluar el desempeño del modelo de simulación basado en agentes utilizando un estudio de caso del cuidado de salud domiciliaria.

2.3. ALCANCE

El alcance de éste proyecto de investigación es la construcción de un modelo de simulación basado en agentes para resolver el problema de cuidado domiciliario mediante el desarrollo del código computacional necesario para ello. Adicionalmente se realizan pruebas preliminares (simulaciones) para evaluar la ejecución del modelo de simulación basado en agentes y por último se realiza un caso de estudio de un sistema de cuidado domiciliario en la ciudad de Bogotá.

2.4. JUSTIFICACIÓN

El cuidado de la salud domiciliar surge con el fin de disminuir costos generales en los sistemas de salud, para mejorar la utilización de recursos que cada vez se hacen más escasos y también sirve para mejorar el servicio a personas vulneradas o con discapacidades físicas entre otras. El hecho de que recursos como el financiero o el talento humano se vuelvan cada día más escasos se da por distintos factores sociales y económicos.

Un sistema básico de cuidado a la salud domiciliar puede contener los siguientes elementos o en el contexto de la metodología por agentes, agentes autónomos: El paciente (agente tarea), el Operador de salud (agente recurso) y la institución prestadora del servicio (agente coordinador). Las interacciones vienen dadas por los objetivos individuales de cada agente y también por el objetivo común del sistema que es la prestación del servicio. Un ejemplo de ello es la interacción entre el agente coordinador y el agente recurso, la cual viene dada por la programación de tareas de los agentes recursos por parte del agente coordinador; otra interacción es la del agente tarea (paciente) y el agente coordinador (institución) la cual se da por una demanda de servicios médicos por parte del paciente que acude al agente coordinador para satisfacer su demanda. Finalmente la interacción entre el agente tarea y agente recurso viene dada en la prestación directa del servicio de salud siguiendo una programación ya establecida.

Mediante el modelo de simulación basado en agentes propuesto se pretende realizar una mejor programación y un ruteo de operadores de salud que la que se realiza actualmente la cual es hecha manualmente. Lo anterior debido a que los proveedores de salud cuentan con limitadas o ninguna herramienta que apoye el proceso de toma de decisiones específicamente a nivel operativo.

2.5. METODOLOGÍA

La metodología a seguir en el proyecto de investigación es la desarrollada por el grupo de investigación Sistemas Expertos y simulación. Con la particularidad de que la metodología del proyecto se enfocará en la modelación y simulación de sistemas basados en agentes. En la Figura 1 se muestra la metodología mediante un diagrama.

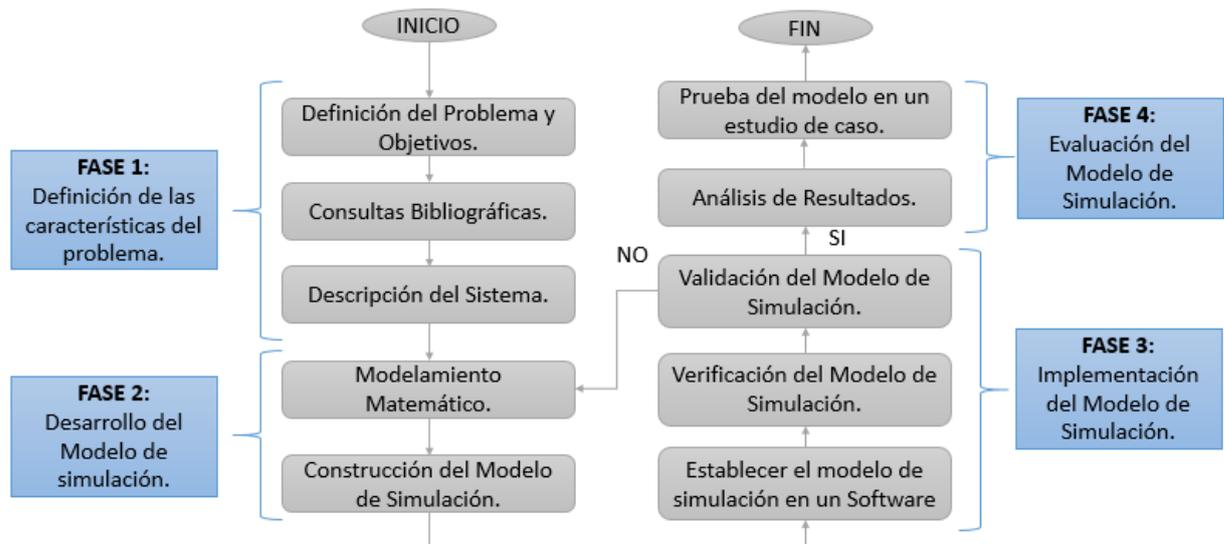


Figura 1 Metodología del proyecto de investigación
Fuente: Autor

Finalmente en la Figura 2 se muestra más detalladamente la estructura metodológica de la investigación con base en su desarrollo por capítulos.

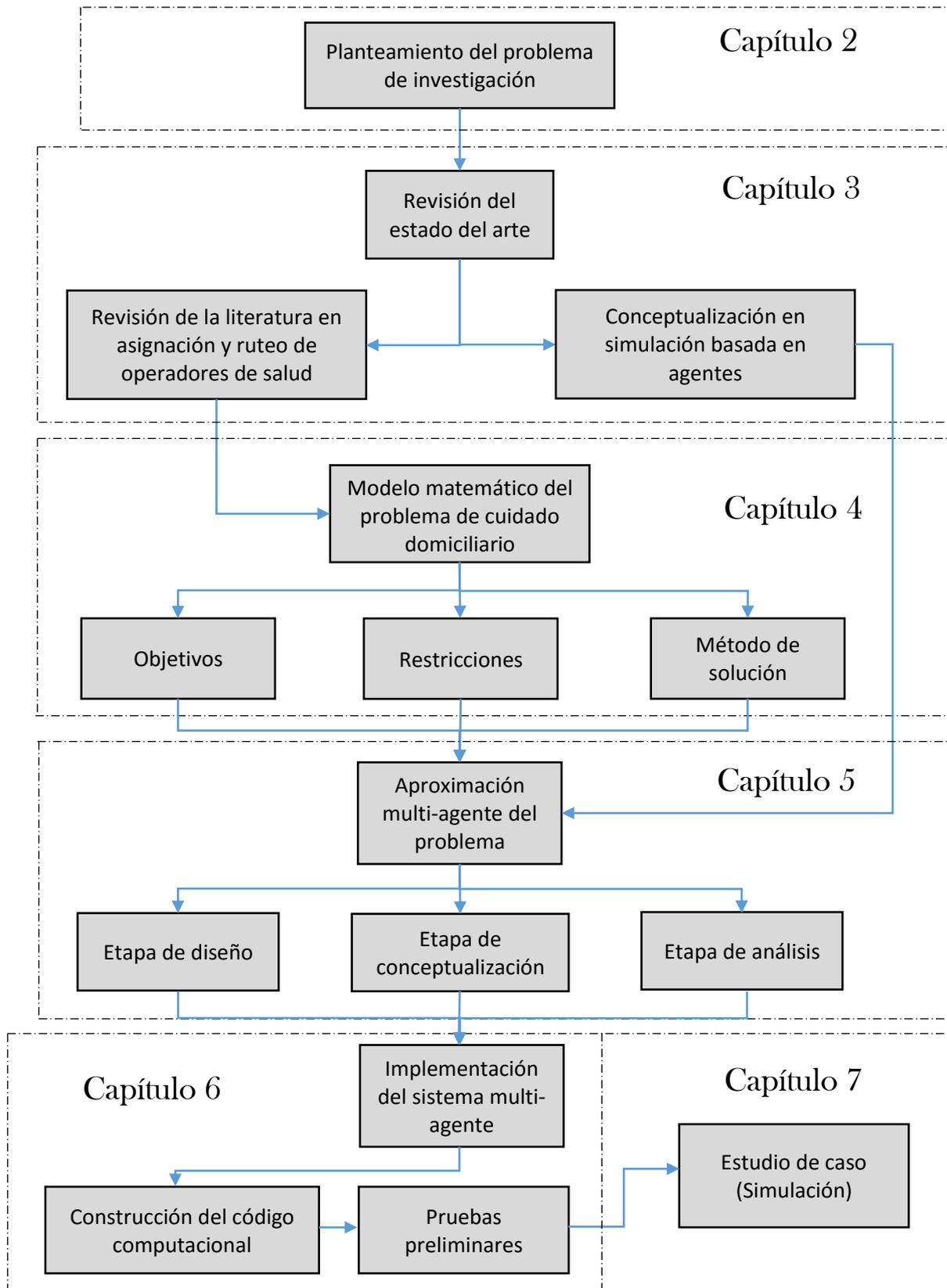


Figura 2 Diagrama de flujo de la estructura metodológica de la investigación
Fuente: Autor

Capítulo 3

REVISIÓN DEL ESTADO DEL ARTE

3.1. REVISIÓN DE LA LITERATURA EN ASIGNACIÓN Y RUTEO DE OPERADORES DE SALUD EN EL CUIDADO DE LA SALUD DOMICILIARIA

Los problemas de asignación y de ruteo de operadores de salud son los grandes componentes de la programación del cuidado a la salud domiciliaria a nivel operativo. La investigación de operaciones (de ahora en adelante IO) ha sido una herramienta que ha sido utilizada desde las últimas décadas para apoyar la toma de decisiones a la hora de realizar programación en estos sistemas. Algunos de los primeros autores que abordaron el problema de asignación y ruteo de personal de salud desde la IO fueron Begur, Miller & Weaver [12], los cuales mediante la utilización de heurísticas y otras técnicas propias de la IO desarrollaron un sistema que lograba asignar y hacer el ruteo de cuidadores. Cheng & Richs [13] dieron tempranos aportes desde el campo de la IO proponiendo un modelo de programación matemática para resolver el problema de asignación y ruteo de enfermeros. Desde ése entonces el interés en el campo de la IO para resolver problemas del cuidado domiciliario ha venido creciendo rápidamente. Inclusive se ha llegado a afirmar que el problema de asignación y ruteo en el cuidado domiciliario puede ser visto como una mezcla de dos problemas característicos de la IO, como lo son el problema de asignación de personal conocido como **Staff Rostering Problem (SRP)** y el problema de ruteo de vehículos conocido como **Vehicle Routing Problem (VRP)** [14]. Otros autores como Kergosien, Lenté & Billaut [15] abordan la asignación y ruteo de operadores de salud como una extensión del problema del agente viajero conocido como **Traveling Salesman Problem (TSP)**. En [10] se realiza una revisión de la literatura que trata la asignación y ruteo en el cuidado domiciliario como variantes del VRP y TSP antes señalados.

Se utilizaron diferentes criterios para realizar la revisión literaria del cuidado domiciliario los cuales son: (i) componentes del problema, debido a que existe literatura que solo tiene en cuenta la asignación de cuidadores sin contemplar el ruteo de ellos, (ii) Naturaleza del sistema, ya que es posible tener en cuenta la incertidumbre en el cuidado domiciliario, (iii) objetivos del sistema, los cuales definen un sistema de cuidado domiciliario, (iv) restricciones, las cuales confieren las características propias del cuidado domiciliario, (v) métodos de solución, que es la manera que los autores abordan el problema para llegar a una solución, (vi) datos de prueba, los cuales pueden representar un estudio de caso real o no. Con base en los anteriores criterios se realizó una clasificación consolidada de las referencias revisadas la cual se encuentra en el Anexo 1. Adicionalmente se realizaron consideraciones sobre la planeación dinámica en el cuidado domiciliario así como del software específico que es utilizado en la actualidad para realizar la planeación operativa del cuidado domiciliario.

3.1.1. Metodología para la revisión del estado del arte

La revisión literaria se realizó con base en la consulta de bases de datos especializadas específicamente del área de ingeniería (IEEE, SPRINGER LINK, SCIENCE DIRECT, SCOPUS). En la búsqueda se utilizaron las palabras clave **Home Health Care Routing, Home Health Care Scheduling, Care at home routing, Care at home scheduling**. Se buscó literatura del tema preferiblemente con una antigüedad no superior a 10 años, sin embargo se incluyó alguna literatura previa por su significativo aporte al tema. De los escritos consultados se identificaron los objetivos, métodos y técnicas empleadas además de las conclusiones a las que los autores llegaron.

Posteriormente se realizó una clasificación o sistematización de la literatura según los objetivos, medidas de desempeño, restricciones, métodos de solución utilizados, la procedencia de los datos de prueba (reales o generados) y si se consideraba la incertidumbre en los trabajos realizados. Además de lo anterior se identificaron algunas herramientas computacionales que son utilizadas en la actualidad para realizar la asignación y el ruteo de operadores de salud, así como si la planeación es realizada de manera dinámica (en tiempo real) o no. Con base en el trabajo anterior se elaboraron algunas conclusiones y se identificaron tendencias y trabajos futuros en el tema.

3.1.2. Problemas de asignación vs problemas de asignación y ruteo

En la literatura correspondiente a la planeación operativa del cuidado a la salud domiciliaria existen trabajos que pueden abordar uno o ambos componentes del problema. Es decir, existe literatura que sólo tiene en cuenta a la asignación sin tener en cuenta el componente de ruteo así como existe literatura que aborda ambos componentes. En trabajos como [16], [17] sólo se tiene en cuenta la asignación de tareas a los profesionales de la salud sin tener en cuenta su ruteo. Estos trabajos se centran solo en determinar la secuencia de tareas, trabajos o los pacientes que deben visitar los operadores de salud sin considerar los costos ni los tiempos de viaje para realizar las visitas.

Por otra parte existen trabajos que han abordado ambos componentes del cuidado a la salud domiciliaria. Para abordar ambos componentes se puede optar primero por realizar la asignación de tareas para luego encontrar las mejores rutas a seguir. Issaoui, Zidi, Marcon & Ghedira [18] abordan el problema en tres fases. La primera fase resuelve el problema de asignación, la segunda construye las rutas para cada cuidador y la tercera fase trata de mejorar la solución encontrada en la segunda fase. Otro trabajo en el cual se descompone el problema de asignación y ruteo en fases se encuentra en [19]. La manera más común de abordar ambos componentes es hacerlo simultáneamente. Técnicas como generación de patrones encontrada en [20] logran tratar el problema sin descomponerlo en fases. Robustos modelos de programación matemática también logran tratar el problema como una sola unidad, ejemplo de ello es el modelo de programación matemática encontrado en [21] que es resuelto mediante la utilización de heurísticas. En la Tabla 1 se hace una clasificación de la literatura revisada según el componente abordado.

Tabla 1 Clasificación de referencias revisadas según componente abordado

Fuente: Autor

MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA

Componente	Referencias revisadas
Asignación	[16, 17, 24, 32, 53, 66, 67]
Asignación y ruteo	[1, 5, 6, 14, 15, 18-23, 25-31, 33-42, 44, 47-49, 54-58, 60, 61, 71, 72, 76, 77]

3.1.3. Problemas estocásticos vs problemas determinísticos

Los problemas de asignación y ruteo de operadores de salud pueden considerar la incertidumbre en las operaciones propias del cuidado a la salud domiciliaria. Tales problemas que incorporan la incertidumbre se clasifican en el tipo estocástico. En [22] se consideran tiempos variables en la prestación del servicio de salud, es decir, que el tiempo que gasta un operador de salud en atender un paciente no es determinístico sino que varía en según una función de densidad de probabilidad. Los autores realizan un modelo de programación estocástica para resolver el problema y lo solucionan mediante técnicas como lo son generación de patrones entre otras. Otro trabajo realizado en el cual se consideró la incertidumbre es el encontrado en [23]. Los autores incluyen incertidumbre tanto en los tiempos de viaje del personal de salud, en el tiempo de las visitas domiciliarias y en el número de visitas que cada paciente requiere.

En ese orden de ideas, la clasificación restante son los problemas que no incorporan incertidumbre en el problema, es decir, los problemas determinísticos debido a que manejan los parámetros de manera determinística. En la Tabla 2 se encuentran la clasificación de las referencias revisadas que son de naturaleza determinística.

Tabla 2 Clasificación de referencias revisadas según consideración de la incertidumbre

Fuente: Autor

Tipo	Referencias revisadas
Determinístico	[5, 6, 14-21, 23, 25-31, 33-42, 44, 45, 47, 49, 52-58, 60, 61, 71, 72, 76, 77]
Estocástico	[1, 22-24, 33, 66, 67]

3.1.4. Funciones objetivo y/o medidas de desempeño

Las funciones objetivo y/o medidas de desempeño son utilizadas para relacionar las variables de decisión del sistema, medir el comportamiento de éste, tener un referente para saber cuál solución es mejor que otra y comparar la solución obtenida por lo propuesto con la solución actual del sistema. Es por lo anterior que en la planeación de nivel operativo del cuidado a la salud domiciliaria se presentan funciones objetivo o medidas de desempeño comunes como las que se analizarán a continuación. Es importante resaltar que en la Tabla 3 se encuentra una clasificación de la literatura revisada en función de las diferentes funciones objetivo tenidas en cuenta.

Un objetivo común en sistemas de cuidado domiciliario es la **minimización de los tiempos** de viaje o tiempos de espera de los operadores de salud así como los tiempos de espera de los pacientes para recibir su atención domiciliaria. En [24] se utiliza el tiempo como medida de desempeño al evaluar su sistema multi-agente observando el tiempo promedio de espera de los pacientes, así como la utilización de recursos para escoger una solución al problema. Otra investigación en la cual se utiliza el tiempo como función objetivo o medida de desempeño se encuentra en [25]. Los autores en su modelo de programación matemática buscan minimizar los tiempos de viaje y los tiempos de espera de los médicos o los operadores de salud.

Otro objetivo en sistemas de cuidado domiciliario es la **minimización del costo** de la prestación del servicio. El costo puede tener uno o varios componentes dependiendo del sistema. Por una parte están los costos de viaje o de transporte que se pueden conocer calculando el costo por kilómetro o por tiempo recorrido. Por otra parte se puede considerar el costo acarreado por las demoras o la no prestación del servicio. Lo anterior se realiza imponiendo penalizaciones monetarias por incumplimiento en el servicio. Costos de prestación del servicio como los de las horas regulares y horas extras trabajadas por los operadores de salud y los costos por subcontratación del servicio también pueden ser considerados. En [22] además de tener en cuenta el costo del viaje de los cuidadores, imponen penalidades monetarias a las demoras en el arribo de los operadores de salud a los hogares de los pacientes. En [17], [26] se consideran los costos de la prestación del servicio como horas extra y subcontratación.

Por otra parte la **minimización de la distancia recorrida** por los operadores de salud también puede representar un objetivo del sistema de cuidado a la salud domiciliaria. Para ello es necesario conocer la ubicación de los pacientes o en su defecto conocer las distancias entre sus ubicaciones y si es necesario entre los pacientes y la institución prestadora del servicio. En el algoritmo basado en optimización de enjambre propuesto por Akjilratikar, Yenradee & Drake [27] la función de bondad para el algoritmo es la distancia. En [28] se busca entre otros objetivos la minimización de la distancia recorrida por los operadores de salud.

La maximización del balance de trabajo entre los operadores de salud también puede convertirse en un objetivo en el cuidado domiciliario. Esta función objetivo aplica principalmente en sistemas en donde existe un número limitado o escaso de operadores de salud y se deba buscar mejorar sus condiciones laborales. Aspectos como cumplir con la legislación de horas laborales, la distribución equitativa en términos de números de pacientes a atender entre los cuidadores así como respetar los tiempos de almuerzo entre otros hacen parte de éste objetivo del sistema. En [16] los autores buscan como objetivo cumplir las horas de trabajo para cada turno de los operadores de salud así como lograr un balance de trabajo justo entre ellos. Los autores En-nahli, Allaoui, & Nouaouri [29] proponen un modelo de programación matemática que dentro de las múltiples funciones objetivo a cumplir se encuentra lograr un equilibrio justo entre la asignación de pacientes a los operadores de salud así como equilibrio en los tiempos totales de consulta por operador de salud.

Lograr que los pacientes estén satisfechos con la prestación del servicio también representa un objetivo en distintas investigaciones sobre el tema. Respetar los horarios de visita determinados por los pacientes (ventanas de tiempo), asignar la visita domiciliaria a un operador de salud suficientemente calificado, tener en cuenta el estado médico del paciente y sus limitaciones físicas, el idioma hablado por el operador de salud, que un solo operador de salud sea el encargado de realizar todas las visitas de determinado paciente (continuidad del cuidado) ,entre otras, son preferencias que hay que tener en cuenta a la hora de prestar un buen servicio de cuidado a la salud domiciliario [30]. Para investigadores como Nickel, Schröder & Steeg [31] la satisfacción de los pacientes es de vital importancia, por lo cual en su modelo matemático incorporan un indicador de lealtad entre paciente y operador de salud, con el fin de asegurar la continuidad del cuidado. Carello & Lanzarone [32] realizan un modelo de programación matemática en el cual el objetivo es la compatibilidad paciente-operador de salud así como la continuidad del cuidado, **maximizando la satisfacción de los pacientes.**

Otra función objetivo tenida en cuenta en el sistema es la **minimización de las violaciones a las restricciones.** Las restricciones en el sistema se utilizan para acercar a la realidad el modelo de programación matemática. En el contexto de la programación matemática es posible considerar restricciones duras y restricciones blandas [33], haciendo énfasis en que las de tipo duro no ser violadas mientras que las de tipo blando pueden ser quebrantadas acarreando una penalización. El trabajo realizado por [6] contempla una programación por restricciones las cuales son de tipo duro y blando. Los autores buscaron la minimización del incumplimiento de las restricciones acarreando un costo o penalización.

Minimizar cantidad de operarios de salud requeridos también puede ser un objetivo del sistema. Para lograr lo anterior se deber realizar una asignación y ruteo que atendiendo todos los pacientes y cumpliendo las restricciones permita que el personal contratado sea el menor posible generando menores costos en el sistema. En [34] los autores proponen un modelo de programación matemática en el cual buscan minimizar una variable binaria que define si un operador de salud es necesario en la asignación y ruteo o no.

En el caso de que en un sistema de cuidado a la salud domiciliaria busque cumplir más de un objetivo a la vez se debe recurrir a la **programación multi-objetivo.** Generalmente cuando el planeador considera con base en su experticia que una función objetivo es más importante para el sistema que las otras le asigna mayor peso o ponderación a ésta. Un ejemplo de lo anterior se encuentra en [28], en donde se persiguen tres objetivos o metas los cuales son la lealtad del paciente-cuidador, costos laborales de los enfermeros y la distancia recorrida en el sistema. Mediante parámetros identificados con la letra α los autores realizan la ponderación de las funciones objetivo. Otro ejemplo de programación multi-objetivo se presenta en [21] en donde se realiza un análisis entre el objetivo de minimizar costos de operación y el objetivo de maximizar la satisfacción del cliente. En [35] se busca minimizar los tiempos de consecución de las actividades de los operadores de salud (*makespan*) y el tiempo computacional en que se resuelve el problema.

Tabla 3 Clasificación de referencias revisadas según la función objetivo

Fuente: Autor

Función objetivo	Referencias revisadas
Minimización de tiempo (viaje - espera)	[6, 14, 23-25, 29, 35, 36, 39, 41, 42, 44, 45, 47, 54, 57, 77]
Minimización de costos operacionales	[5, 14, 15, 17, 18, 22, 26, 28, 29, 32, 33, 37, 38, 48, 49, 52, 55, 67, 72, 76]
Minimización de distancia recorrida por los operadores de salud	[1, 19, 20, 27, 28, 30, 31, 58, 60, 61]
Maximización del balance de trabajo entre los operadores de salud	[6, 16, 19, 29, 33, 40, 44, 49, 53, 56, 66]
Maximización de la satisfacción de los pacientes	[18, 21, 22, 28, 29, 30-32, 38, 42, 45]
Minimización de las violaciones de las restricciones	[6, 40, 57, 60, 66, 77]
Minimización del número de operadores de salud	[34, 56, 71, 72]
Múltiples objetivos	[6, 14, 17, 18, 21, 22, 25, 28-33, 35, 38, 40, 42, 44, 45, 53, 56, 57, 60, 66, 72, 77]

3.1.5. Restricciones del problema

A la hora de realizar la planeación operativa del cuidado a la salud domiciliario se deben tener en cuenta una serie de requisitos para brindar el servicio. Horarios en los que los pacientes deben ser atendidos, habilidades y capacidades requeridas por los pacientes en sus visitas, la necesidad de que uno o más operadores de salud lleguen al tiempo para realizar la visita, entre otras son características que definen un sistema como uno propio del cuidado domiciliario. La manera de tener en cuenta esas características del sistema es mediante restricciones o decisiones de planeación. A continuación se revisarán las más comunes en los trabajos realizados en el tema y en la Tabla 4 se encuentra la clasificación de referencias revisadas según sus restricciones.

Una restricción muy común en sistemas de cuidado domiciliario son las **ventanas de tiempo**. Estas surgen debido a que los pacientes que reciben servicios del cuidado a la salud domiciliaria pueden no estar disponibles en todo momento para recibir las visitas. Es por lo anterior que los pacientes generalmente determinan franjas horarias para tomar las visitas de los operadores de salud. Estas franjas horarias son conocidas como ventanas de tiempo y son frecuentemente consideradas en la literatura disponible del tema. Algunos de los trabajos que han tomado en cuenta las ventanas de tiempo de los pacientes son [27], [33], [36]-[39].

Otra restricción común del sistema corresponde a las **habilidades de los operadores de salud**. Las instituciones proveedoras del cuidado domiciliario deben garantizar que el profesional de la salud que envíen a realizar las visitas esté suficientemente calificado para realizarlas. En consecuencia se deben clasificar las habilidades de los operadores de salud así como los requerimientos de los pacientes. En [40] utilizan matrices de compatibilidad entre pacientes y operadores de salud para decidir que operador de salud está calificado

para atender a determinado paciente. Otra manera de tener en cuenta las habilidades de los operadores de salud es la realizada en [41]. Los autores determinan el nivel de calificación del operador de salud y el nivel de requerimiento de la visita domiciliar a determinado paciente. Con los parámetros anteriores en su modelo matemático buscan que el operador de salud realice la visita domiciliar siempre que su nivel de habilidad sea mayor o igual que el nivel requerido por el paciente.

Tanto los pacientes como los operadores de salud pueden tener ciertas razones o **preferencias** que les impida tomar o realizar visitas domiciliarias respectivamente. En [6] los autores consideran y agrupan en conjuntos las razones para no recibir o tomar una visita domiciliar. Por ejemplo, en el caso de que un operador de salud que tenga alergia a determinada mascota no podría visitar a pacientes que convivan con esa mascota. En el caso de los pacientes pueden preferir ser atendidos por un operador de salud mujer. Lo anterior es manejado por los autores en notación de conjuntos y buscando la intersección de éstos conjuntos se asigna o no el paciente al operador de salud. Otra forma de tener en cuenta las preferencias se encuentra en [42], en donde mediante un parámetro binario se establece si un paciente prefiere ser atendido por un operador de salud determinado o no.

Los profesionales médicos que trabajan en sistemas de cuidado domiciliario no pueden disponer la totalidad de su tiempo a la institución que brinda éste servicio. Ellos pueden tener diferentes ocupaciones u obligaciones en el día como por ejemplo tareas del hogar y además requieren de tiempo de descanso para poder cumplir sus labores de una mejor manera. **El tiempo de trabajo de los operadores de salud** se determina mediante negociaciones entre la institución prestadora del servicio y el profesional médico o mediante normatividad laboral. Para tener en cuenta ésta restricción los autores ponen límites superiores del tiempo de servicio del operador de salud en el sistema. En [20] se establece un parámetro que indica el máximo tiempo de trabajo diario permitido por los operadores de salud. En [30] también se manejan parámetros para limitar el tiempo de trabajo máximo, los autores establecen mediante un parámetro el tiempo de trabajo máximo por semana de los operadores de salud y lo tienen en cuenta en su modelo de optimización. En [5] tienen en cuenta el tiempo de trabajo de los operadores de salud mediante ventanas de tiempo duras las cuales determinan el inicio y el final de los turnos. En [19] se tiene en cuenta el tiempo de trabajo de mediante un parámetro que impone un tiempo de trabajo máximo teniendo en cuenta los tiempos de viaje así como de servicio de los operadores de salud.

La **precedencia** en el cuidado domiciliario supone que para realizar determinada visita domiciliar se debió realizar antes otra visita. En [38] se trata este tema ajustando las ventanas de tiempo de las visitas para lograr que unas sean realizadas obligatoriamente antes que otras. En [25] es propuesto un modelo de programación matemática que utiliza un parámetro binario el cual indica si el operador de salud visita al paciente antes de otro operador de salud. Chahed, Marcon, Sahin, Feillet & Dallery [43] realizan un gran aporte al estudio del cuidado domiciliario tratando el tema de la atención de pacientes que requieren quimioterapia en casa. Para ello los autores en su modelo matemático consideran la producción y la entrega de drogas para éstos pacientes, teniendo en cuenta que cierta droga no puede ser producida antes que otra, es decir, tienen en cuenta la precedencia en su aproximación al tema.

La **sincronización** considera que dos o más operadores de salud estén al mismo tiempo realizando una visita domiciliaria. Existen varios procedimientos médicos que exigen que intervengan varios profesionales simultáneamente. Desde la visita de un médico que requiera apoyo de enfermería, la realización de terapias especializadas hasta el levantamiento de un paciente pesado puede requerir el apoyo de más de un operador de salud. Lo anterior es tenido en cuenta en [44] mediante un parámetro que determina el número de operadores de salud requeridos por actividad. En [45] se trata el tema de sincronización mediante un parámetro que determina el tiempo entre visitas domiciliarias para un paciente, si el paciente requiere que dos o más operadores de salud estén al tiempo realizando su visita domiciliaria este parámetro se reduce a cero. En el modelo de programación matemática propuesto en [46] incorpora la sincronización ajustando para los operadores requeridos por el paciente la variable que determina su inicio de la visita domiciliaria en el mismo paciente.

Otro aspecto que es considerado en la asignación y ruteo de operadores de salud es el **punto de partida y finalización de sus recorridos**. Lo anterior puede ocurrir ya sea en las instalaciones del centro de salud o en el hogar de los operadores de salud. En [35] se propone un modelo matemático en el cual los operadores de salud deben obligatoriamente comenzar y finalizar sus recorridos desde centro de salud que hace las veces de depósito. Por otra parte en [47] se cuenta con información de la ubicación de los hogares de los pacientes los cuales corresponden a los nodos de partida y de llegada de los operadores de salud. Otro aporte significativo en el tema es el hecho por Bard, Shao, & Wang [48]. Los autores en su aproximación al tema mediante distintos modelos matemáticos consideran por separado el origen tanto en las instalaciones del centro de salud como en los hogares de los operadores de salud.

Tabla 4 Clasificación de las referencias revisadas según sus restricciones

Fuente: Autor

Restricción	Referencias revisadas
Ventanas de tiempo	[5, 6, 14-16, 18, 21-23, 25-31, 33-42, 44, 45, 47-49, 54, 55, 57, 58, 60, 61, 66, 76, 77]
Habilidades de los operadores de salud	[1, 5, 6, 15-20, 26, 29, 31-34, 41, 42, 44, 47, 49, 52, 53, 57, 58, 61, 66, 71, 76, 77]
Preferencias de los pacientes -continuidad del cuidado	[1, 6, 15, 17, 26, 32, 37, 49, 42, 57, 71]
Tiempo de trabajo de los operadores de salud	[1, 5, 6, 14-21, 23, 25, 26, 28, 30-34, 36, 40-42, 47, 48, 52, 53, 56, 57, 66, 67, 71, 72, 77]
Precedencia de las visitas	[15, 25 36-38, 45, 55, 60]
Sincronización de los operadores de salud en las visitas	[15, 35, 40, 44, 45, 52, 53, 55, 76]
Punto de partida y finalización de los	Un depósito - centro de salud [1, 5, 19-22, 25, 29, 31, 33, 35, 36, 39-42, 44, 48,52, 54, 55, 57, 58, 60, 61]

MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA

operadores de salud (depósito)	Múltiples depósitos - hogar de los operadores	[6, 14, 15, 23, 26, 27, 30, 34, 37, 38, 45, 47, 48, 49, 57]
	No especificado	[16-18, 24, 28, 32, 53, 56, 66, 67, 71, 72, 76, 77]

3.1.6. Métodos de solución

La investigación de operaciones es una gran caja de herramientas, métodos, técnicas y algoritmos para resolver problemas de optimización [49]. El objetivo de los métodos de optimización es encontrar la mejor solución al problema o una solución cercana de ésta utilizando un el menor esfuerzo computacional posible. Se pueden distinguir dos tipos de métodos de solución. Los métodos exactos que garantizan encontrar una solución óptima y los métodos aproximados que no garantizan encontrar la mejor solución, sin embargo tienen mejor desempeño en términos de tiempo de solución y utilización de esfuerzo computacional. Generalmente se escogen métodos exactos cuando los problemas pertenecen a la clase P, es decir, el esfuerzo computacional crece de manera polinómica. Cuando los problemas pertenecen a la clase NP-Completo, es decir, requieren esfuerzos computacionales exponenciales generalmente son abordados utilizando métodos aproximados [50]. Es importante resaltar que el problema de asignación y ruteo es de tipo NP-Completo [25], [34], [49].

Se destaca que en mayoría de los trabajos revisados los autores en una primera instancia utilizan métodos solución exactos como lo son la programación lineal entera o la programación por restricciones. Lo anterior es realizado con el fin de resolver problemas de pequeña magnitud (pocos pacientes y pocos operadores de salud). En una segunda instancia los autores desarrollan métodos aproximados para abordar problemas de mayor envergadura como las que se encuentran en las aplicaciones reales [51]. A continuación se revisarán los diferentes métodos que han sido utilizados para resolver el problema del cuidado domiciliario a nivel operativo.

Métodos exactos

Los métodos exactos tienen la característica de realizar búsquedas exhaustivas en el espacio de soluciones. Al realizar ese tipo de búsqueda se asegura obtener la solución óptima del problema. Dentro de los métodos exactos se encuentran los algoritmos de ramificación y corte como los utilizados en [26], [38], los cuales descomponen el problema en ramas y van podando o eliminando las que no son óptimas para el problema. En [20], [22], [52], se utiliza la generación de columnas como método para solucionar el problema de asignación y ruteo de operadores de salud. Otro método exacto utilizado ha sido la programación lineal entera mixta como se evidencia en [16], [18], [23], [39], [44], [53]. La programación dinámica también ha sido utilizada para abordar éste tipo de problemas como se evidencia en [41]. En la Tabla 5 se encuentra una clasificación de las referencias revisadas según el método exacto utilizado.

Por otra parte distintos programas computacionales han sido utilizados para resolver el problema de programación y ruteo de operadores de salud utilizando métodos exactos. En [53], [54] se utilizó el software ILOG CPLEX para obtener soluciones al problema. En [37], [42] se ejecutó el modelo de programación matemática en el software XPRESS. Otro lenguaje que ha sido utilizado para ejecutar modelos de programación matemática es Java. En [30] se utiliza Java junto con ILOG CPLEX para implementar y ejecutar el método exacto que permite llegar a la solución del problema.

Tabla 5 Clasificación de las referencias revisadas según el método exacto utilizado

Fuente: Autor

Método exacto	Referencias revisadas
Programación lineal entera mixta	[1, 16, 17, 25, 29 39, 44, 47, 53, 54, 71, 72]
Algoritmos de ramificación y corte	[26, 38]
Programación dinámica	[41]
Generación de patrones	[19, 20, 34, 52]

Métodos aproximados

Los métodos aproximados se caracterizan por buscar soluciones de distintas maneras en el espacio de soluciones del problema. Existen métodos aproximados que son inspirados por la naturaleza. En el caso del cuidado a la salud domiciliario los algoritmos genéticos utilizados en [55], los algoritmos evolutivos utilizados en [56], la búsqueda tabú utilizada en [49], [55], [57]–[59], el recocido simulado utilizado en [6] son heurísticas o metaheurísticas que tienen la característica de basarse en procesos naturales para buscar soluciones. Existen también métodos aproximados que se basan en algoritmos de búsqueda local o de vecindario, son encontradas en [21], [31], [42], [60], [61] que fueron utilizados para encontrar soluciones al problema. En la Tabla 6 se encuentra la clasificación de las referencias revisadas según el método aproximado utilizado.

Generalmente éstos métodos son implementados y ejecutados en lenguajes de programación de propósito general como se encuentra en [34], [58] en donde se utilizó C++. Otro lenguaje de programación utilizado es Matlab [27].

Tabla 6 Clasificación de referencias según el método aproximada utilizado

Fuente: Autor

Métodos aproximados	Referencias revisadas
Búsqueda tabú	[37, 41, 49, 55, 57, 58, 60, 77]
Recocido simulado	[6, 77]
Algoritmos de búsqueda	[6, 14, 18, 28, 30, 31, 42, 60, 61]
Optimización de enjambre	[27, 57, 66]
Heurísticas desarrolladas por los autores	[33-36]
Algoritmos genéticos	[55]
Heurística greedy	[5, 14, 45]

Otros métodos de solución

Los sistemas multi-agente o los modelos basados en agentes han sido utilizados durante los últimos años para realizar el ruteo de vehículos [62] así como para realizar programación de tareas en diferentes industrias ya sea de manufactura como de servicios [63], [64]. Al explotar las características de los agentes (autonomía, cooperación y coordinación) se puede realizar planeación en tiempo real, es decir, se pueden obtener soluciones que pueden ser alteradas al ingresar nuevas demandas al sistema. Lo anterior es realizado en tiempos de computación razonables y la calidad de las soluciones alcanzadas por éste método es comparable con las soluciones arrojadas por los métodos tradicionales [65]. En la revisión literaria realizada no se encontraron aplicaciones de éste método específicamente para los problemas de cuidado a la salud domiciliario.

La lógica difusa también ha sido utilizada para realizar la programación de tareas en el contexto del cuidado a la salud domiciliario como se evidencia en [40], [66]. Por otra parte, las cadenas de Markov también han sido utilizadas para resolver problemas propios del cuidado a la salud domiciliaria. Koeleman, Bhulai, & Van Meersbergen [67] realizan la programación de personal de un proveedor de cuidado domiciliario utilizando procesos de decisión Markovianos.

Los sistemas expertos constituyen otro método que puede ser utilizado para realizar procesos de planeación en el entorno del cuidado a la salud domiciliario. Mediante una base de conocimiento, motores de inferencia computacional y una interfaz de usuario los sistemas expertos pueden simular el proceso de aprendizaje, memorización, razonamiento y comunicación de un experto en humano [68]–[70]. Este método se traduce en un robusto programa computacional que puede apoyar las decisiones de programación en ambientes de manufactura así como de servicios [68]. Durante la revisión bibliográfica no se encontraron sistemas expertos que tengan en cuenta específicamente la programación y ruteo de personal de salud. Lo anterior puede constituir una oportunidad de investigación para los interesados en el tema.

3.1.7. Datos de prueba

Con el fin de medir el desempeño de un método de solución propuesto se hace necesario realizar experimentación con información propia del sistema. Ubicación geográfica de los pacientes, número y características de los operadores de salud, demanda del sistema, entre otros hacen parte de la información requerida para probar lo propuesto por los autores. Lo anterior se denomina datos de prueba y en algunos casos ésta información corresponde a la realidad así como pueden ser generados aleatoriamente o estar disponibles en la literatura. En resumen los datos de prueba son la información de entrada del sistema con los cuales lo propuesto por el autor será puesto a prueba.

Contar con información real proveniente de instituciones que brinden el servicio es un aporte muy significativo al trabajo de los autores. En esos casos se puede saber si lo propuesto por mejora el esquema actual de programación y ruteo en términos de costos o en los objetivos del sistema. En [30] se estudió el caso de una organización de Bélgica que brinda el servicio de cuidado domiciliario y mediante lo propuesto se encontró que se mejoraba el servicio reduciendo la distancia recorrida por los operadores de salud y

umentando el nivel de servicio de la organización. En [41] se utilizó información de una localidad de Viena encontrando que mediante lo propuesto por los autores y en diferentes escenarios se lograba disminuir costos en un gran porcentaje. En [71] se estudia el caso de un proveedor de cuidado domiciliario alemán el cual realiza sus procesos de planeación operativa de forma manual, la aproximación desarrollada por los investigadores demuestran que mediante lo propuesto por ellos se mejora el objetivo del sistema que en este caso es la continuidad del cuidado.

En casos en que los autores no dispongan de información real de instituciones que presten el servicio se hace necesario que se genere aleatoriamente la información de entrada del problema. En [25], [72] los autores generaron aleatoriamente los datos de prueba con un número determinado de pacientes y de cuidadores así como los tiempos de viaje entre los pacientes. Como el problema de asignación y ruteo puede ser visto como una variante del problema de ruteo de vehículos con ventanas de tiempo, instancias o datos ya establecidos en la literatura son utilizadas como información de entrada en el sistema. Las instancias de Solomon son utilizadas en [55], [58] con el fin de establecer la ubicación de los pacientes así como las ventanas de tiempo en que pueden ser atendidos. En la Tabla 7 se encuentra una clasificación de las referencias revisadas según los datos de prueba.

Tabla 7 Clasificación de las referencias revisadas según los datos de prueba

Fuente: Autor

Datos	Referencias revisadas
Reales	[1, 5, 6, 14, 17, 19, 20, 23, 26, 27, 30, 31-33, 36-38, 41, 42, 47, 49, 54, 56, 71, 76]
Generados aleatoriamente o encontradas en la literatura	[15, 16, 18, 21, 22, 24, 25, 28, 29, 34, 35, 39, 40, 44, 45, 48, 52, 53, 55, 57, 58, 60, 61, 67, 72, 77]

3.1.8. Consideraciones de planeación dinámica

Durante el desarrollo de las actividades del cuidado domiciliario se pueden presentar imprevistos que requieran que el esquema de asignación y ruteo sea modificado. Cambios en las preferencias de los pacientes, en las ventanas de tiempo de los pacientes o en la disponibilidad de los operadores de salud, así como el ingreso de nuevos pacientes son algunos de los acontecimientos que pueden ser inesperados para el sistema. Lo anterior exige que la planeación sea realizada de forma dinámica permitiendo al sistema responder eficazmente ante éstos imprevistos [14]. En entornos dinámicos como el del cuidado domiciliario la asignación de personal debe ser generada y regenerada regularmente, es decir, los procesos de optimización deben ser llevados a cabo en tiempo real [73]. Un ejemplo de realización de asignación y ruteo realizado de manera dinámica se encuentra en [74], en donde los autores abordan el tema de asignación y ruteo de ambulancias en tiempos de desastres naturales donde estos recursos son escasos. Mediante una metaheurística los autores obtienen soluciones de buena calidad en corto tiempo por lo que puede ser aplicado en entornos dinámicos. En [75] se realiza la asignación y ruteo de ambulancias teniendo en cuenta los tiempos de respuesta de las ambulancias así como las características de incertidumbre y dinámicas de éste entorno mediante reglas de decisión.

3.1.9. Software específico para resolver problemas de cuidado domiciliario

Existen herramientas computacionales que han sido desarrolladas con el fin de apoyar la toma de decisiones específicamente a nivel operativo en el cuidado a la salud domiciliario. Una de éstas herramientas es LAPS CARE la cual tenido un gran impacto en varias organizaciones en donde ha sido utilizada. Esta herramienta computacional cuenta con varios componentes como lo son bases de datos, mapas, rutinas de optimización, entre otros, los cuales le permiten realizar la asignación y ruteo de operadores de salud encontrando mejores soluciones que las que se llega manualmente. Se ha evidenciado que mediante la utilización de ésta herramienta computacional se ha reducido considerablemente el costo operacional del sistema así como tiempo de planeación en diferentes instituciones de cuidado a la salud domiciliaria [76]. PARPAP es otro desarrollo computacional realizado por Bertels & Fahle con el fin de realizar la asignación y ruteo de operadores de salud. Estos autores mediante una combinación de programación lineal, programación por restricciones y diferentes procedimientos heurísticos construyeron el mencionado software [77]. Por otra parte la compañía de software ORTEC tiene un software dedicado exclusivamente para la planeación del cuidado domiciliario. Nickel, Schröder & Steeg [31] comparan los resultados a los que llegaron con su propuesta con los resultados arrojados por dicho software. Finalmente se encuentran librerías dedicadas exclusivamente a la planeación del transporte que han sido utilizadas para resolver los problemas de ruteo en el cuidado domiciliario. La librería para optimización de transporte SPIDER utiliza meta heurísticas y otras técnicas para solucionar problemas de transporte dentro de los que se encuentran los problemas de ruteo en cuidado domiciliario [78].

3.2. CONCEPTUALIZACIÓN DE SIMULACIÓN BASADA EN AGENTES

Según la Real Academia Española (RAE) un agente se define como algo que obra o tiene virtud de obrar y también es definido por la RAE como una persona o cosa que produce un efecto (Real Academia Española, 2001). Una definición más rigurosa de agente es la dada por Russel & Norving [79] que define a un agente como “cualquier cosa capaz de percibir su medio ambiente con la ayuda de sensores y actuar en ése medio utilizando actuadores.”. Otra definición más amplia sobre agentes es la dada por el reconocido autor en el tema Michael Wooldige [80] que define agente como: “un sistema de computadora que está situado en algún entorno o ambiente que es capaz de actuar de forma autónoma e independiente con el fin de cumplir sus objetivos de diseño.”

Un ejemplo básico de un agente interactuando con su entorno es el de un termostato. Mediante sensores el termostato percibe la temperatura de su entorno para luego evaluar la temperatura y regularla autónomamente según sus parámetros de configuración. Los agentes o como algunos autores los llaman agentes inteligentes hacen parte del campo de la inteligencia artificial. El origen de este nuevo paradigma de la computación surge según Michael Wooldige de algunas tendencias de la computación como lo son ubicuidad, interconexión, inteligencia, delegación de tareas y la orientación humana de la interacción con los computadores Michael Wooldige [80].

La simulación basada en agentes comprende la existencia de varios agentes inteligentes interactuando en un mismo entorno dando lugar a los sistemas multi agente. Cada agente actúa a favor del objetivo general del sistema y a la vez busca cumplir sus objetivos de diseño, es decir, cada agente tiene metas y motivaciones propias pero todos los agentes comparten un objetivo general del sistema. Por lo anterior los agentes requieren habilidades de negociación, cooperación y coordinación para cumplir sus propias metas y a la vez estará direccionados en un objetivo global del sistema multi-agente. Además en Macal & North [81] se resalta la interacción entre agente, el aprendizaje de sus experiencias y la adaptación de sus comportamientos para encajar en una mejor forma en su ambiente.

Un claro ejemplo de la utilización de sistemas basados en agentes es la sonda espacial de la NASA Deep Space 1 (DS 1) lanzada a finales de los años 90. La sonda espacial tenía el propósito de ser un demostrador tecnológico. Entre las nuevas tecnologías probadas con la sonda espacial se encontraba un sistema basado en agentes denominado Agente Remoto. Ésta tecnología le dio a la sonda la capacidad de arreglar errores, corregir variaciones menores en los trayectos y la capacidad de tomar decisiones. Lo anterior dio la posibilidad a los operadores en tierra de dar a la sonda los objetivos a cumplir en vez de ordenarle en cada momento exactamente qué hacer y cómo hacerlo. Mediante el sistema Multi Agente la sonda construía y ejecutaba autónomamente la secuencia de operaciones a realizar para alcanzar dichos objetivos [82].

Los investigadores Macal & North [83], contemplan el modelamiento y simulación de sistemas basados en agentes como una técnica que permite modelar sistemas complejos enfocándose en los comportamientos individuales y en la manera en que estos afectan a otros. Los autores también resaltan la variedad de aplicaciones de ésta técnica y la atención que ha ganado en los últimos años. Por ultimo enfatizan la posible aplicación de ésta técnica en las necesidades de los sistemas de salud.

Autores como Davidsson [84] clasifican la simulación basada en agentes como una técnica de simulación micro debido a que modela específicamente comportamientos individuales. Ésta simulación micro emerge de las interacciones de los individuos. Parunak, Savit, & Riolo [85] llegan a una interesante conclusión al indicar que es conveniente utilizar la simulación basada en agentes cuando en el ambiente la unidad de descomposición es el individuo y se puede notar su comportamiento. También resaltan su utilización cuando las interacciones entre los actores del sistema son importantes y es complejo representarlas a través de ecuaciones.

3.2.1. Metodologías para la construcción de modelos de simulación de sistemas basados en agentes.

La simulación basada en agentes es un enfoque relativamente nuevo en la simulación de sistemas en el cual actúan agentes autónomos [86]. Ésta técnica de simulación de sistemas hasta el momento no tiene un procedimiento estándar ni un conjunto de formalismos establecidos para su desarrollo tal como lo tienen otras técnicas de simulación como la dinámica de sistemas [87]. Algunas instituciones han empezado a liderar esfuerzos para estandarizar ésta técnica de simulación, pero una arquitectura de referencia o una metodología completa y estándar para trabajar con agentes aún no existe [88].

En la literatura es posible encontrar una gran variedad de metodologías que dependen de la experticia del investigador. En [87] se encuentra una forma interesante y completa de realizar un modelo de simulación de un sistema basado en agentes. Los autores presentan y profundizan en las diferentes etapas en la construcción del modelo que son presentadas en la Figura 3.

Otro aporte en las metodologías para desarrollar modelos de simulación basados en agentes es la hecha por [89], el autor considera que las metodologías actuales tienen sus fundamentos en diversos campos como lo son la simulación de sistemas, ingeniería de software y la ingeniería de software orientada a agentes. En su trabajo el autor propone una metodología buscando tomar las fortalezas de los campos mencionados anteriormente. La metodología planteada es presentada en la Figura 4.

Por otra parte [90] en su estudio sobre las metodologías para el desarrollo de sistemas Multi agente señala que para la construcción de éstos sistemas se deben integrar tecnologías de distintas áreas del conocimiento. Por una parte se utilizan técnicas de la Ingeniería de Software para darle estructura al proceso. Por otra parte se utilizan técnicas propias de inteligencia artificial para desarrollar la autonomía del sistema y finalmente se utilizan técnicas de programación para la coordinación del sistema por lo que el modelado de sistemas basados en agentes es complejo.

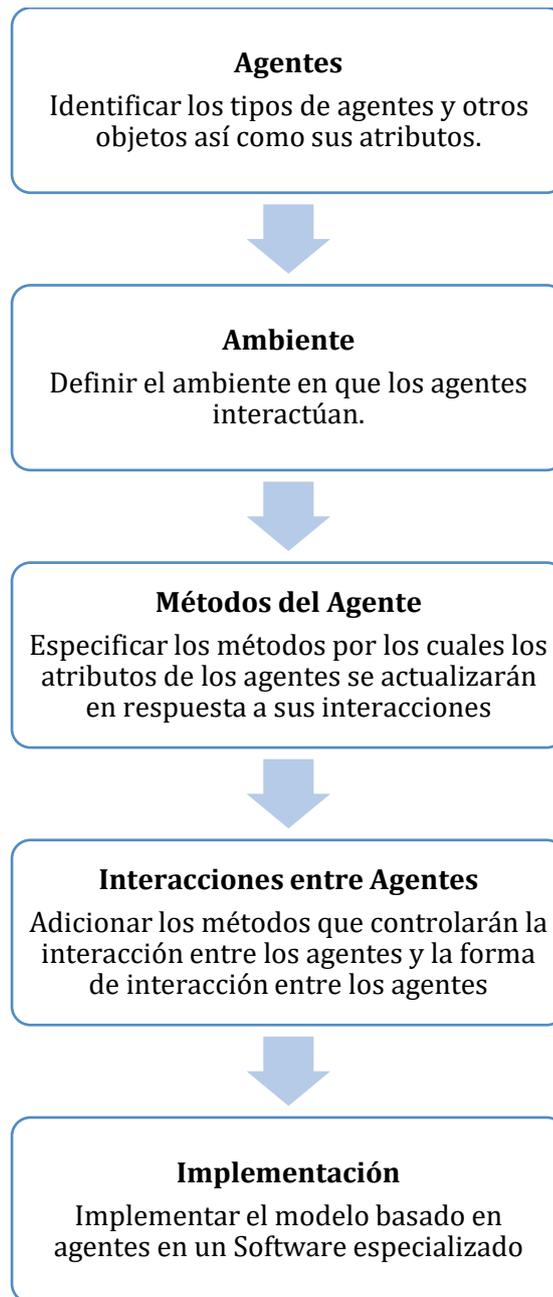


Figura 3 Metodología Simulación basada en Agentes de Macal & North

Fuente: Adaptado de [83]

MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA

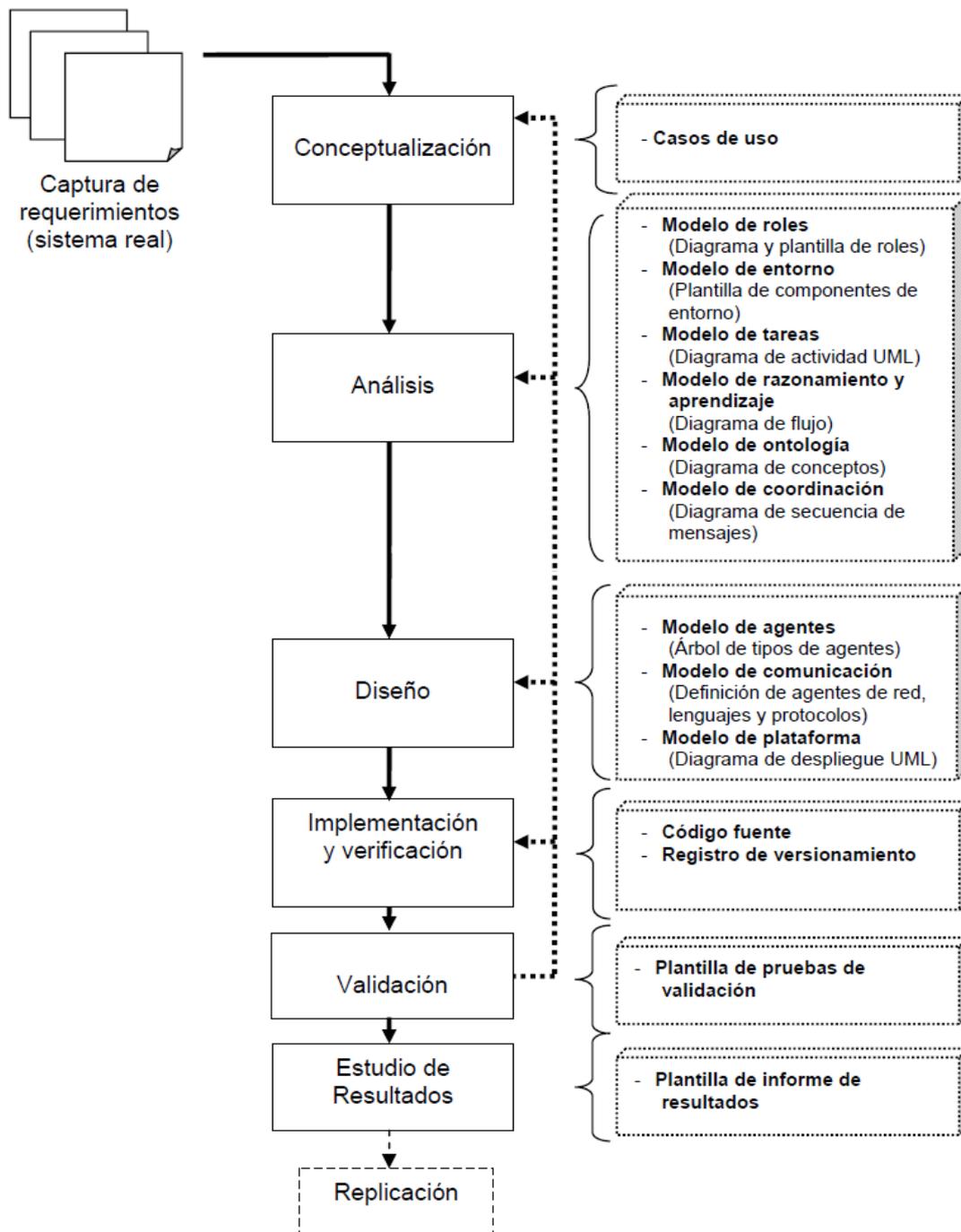


Figura 4 Metodología de simulación de sistemas basados en agente de Moreno

Fuente: Tomado de [89]

3.2.2. Plataformas para el desarrollo de sistemas multi-agente

Existen diferentes plataformas las cuales facilitan el desarrollo de sistemas multi-agente que van más allá de la implementación de la semántica de la programación de los agentes ya que se concentran en aspectos fundamentales como lo son la comunicación y la coordinación de los agentes. En [91] se hace una revisión de los lenguajes de programación y las plataformas con las cuales se desarrollan los sistemas multi-agente. Por otra parte los autores Kalliopi Kravari y Nick Bassiliades [92] realizan una completa y amplia revisión de plataformas utilizadas para construir sistemas multi-agente, a continuación y con base en el trabajo de esos autores se describirán algunas plataformas haciendo énfasis en sus ventajas a la hora de su utilización.

- **JADE (Java Agent DEvelopment Framework):** Es una plataforma para el desarrollo de sistemas multi-agente totalmente implementada en Java que sigue el estándar de lenguaje de comunicación FIPA-ACL, el cual facilita la interacción y comunicación entre los agentes mediante el envío y recepción de mensajes. Jade es de distribución libre y es en la actualidad una de las plataformas más usadas para el desarrollo de aplicaciones o simulaciones multi-agente debido a su amigabilidad con el usuario, su amplia documentación y la disponibilidad de ejemplos sobre su utilización.
- **Jadex:** Plataforma para el desarrollo de sistemas multi-agente que facilita la construcción de los agentes con base en ingeniería de software. Por otra parte permite la programación de agentes inteligentes en XML y en Java. Ha sido utilizado para construir aplicaciones en diferentes dominios como lo son la simulación, programación y computación móvil.
- **AnyLogic:** Es una robusta herramienta computacional la cual además de permitir la simulación basada en agentes también incluye otros métodos de simulación como lo son la simulación de eventos discretos y la dinámica de sistemas. La flexibilidad de su lenguaje de modelamiento permite al usuario capturar la complejidad presentada en sistemas sociales y/o económicos a un gran nivel de detalle.
- **Netlogo:** Es un entorno de programación, modelamiento y simulación multi-agente, el cual puede ser utilizado en sistemas complejos permitiendo dar instrucciones a gran cantidad de agentes los cuales operan independientemente. Es una de las plataformas para el desarrollo multi-agente más utilizadas en la actualidad junto a JADE la cual permite la exploración de fenómenos emergentes, contiene librerías con una gran cantidad de modelos desarrollados de varios campos del conocimiento como lo son química, psicología y dinámica de sistemas.
- **Jason:** Plataforma especializada en el desarrollo de sistemas multi agente la cual es completamente desarrollada en java, por lo que permite personalizar los métodos de los agentes e implementa la semántica de la comunicación que permite establecer la comunicación entre agentes de una manera sencilla.

3.2.3. Revisión de aplicaciones de sistemas multi-agente

Durante los últimos años se han desarrollado distintas investigaciones las cuales han apuntado a la construcción de aplicaciones multi-agente en varios campos del conocimiento. Los sistemas multi-agentes están comprendidos por agentes individuales los cuales se caracterizan principalmente por su autonomía, inteligencia y colaboración entre ellos para cumplir los objetivos del sistema en donde se encuentren [93]. Por lo anterior el paradigma multi-agente resulta prometedor para enfrentar diferentes problemas y existe un gran interés en el desarrollo de modelos basados en agentes para distintos áreas del conocimiento [83].

Un ejemplo de la utilización del paradigma multi-agente se encuentra en [93] en donde los autores logran realizar la programación dinámica de tareas en un sistema de producción tipo Job Shop gracias a la utilización de distintos agentes que interactúan entre ellos. Por otra parte en [94] los autores construyen una plataforma multi-agente con la cual resuelven el problema de ruteo de vehículos de manera dinámica, tal plataforma es implementada utilizando la librería JADE de Java. Otro campo en el cual se ha utilizado el paradigma multi agente es el de la energía [95], los autores resuelven el problema de administración y manejo de energía en redes eléctricas utilizando un para ello un sistema multi-agente.

Por otra parte la logística de servicios de mantenimiento también ha sido abordada utilizando sistemas multi-agente, en [96] los autores construyen un sistema de mantenimiento inteligente el cual es capaz de operar de manera cooperativa y autónoma para hacer un pronóstico de las necesidades de mantenimiento, para mejorar la programación de tareas y para reducir el costo de mantenimiento. En [97] se maneja el paradigma multi-agente para realizar la programación de la producción en manufactura utilizando protocolos de negociación entre agentes para asignar tareas de manufactura entre ellos. La programación de exámenes y asignación de salones utilizando un sistema multi-agente fue realizado en [98] en donde se utiliza un modelo de programación matemática adaptado a un sistema multi-agente. En [99] se realiza un modelo de simulación de una subasta mediante un sistema multi-agente con el fin de calzar ofertas de compradores y vendedores de acciones en la bolsa. En [100] se realiza una simulación de las operaciones de una cadena de suministro utilizando la tecnología de los agentes.

Respecto a aplicaciones multi-agente que abordan problemas relacionados con la salud se encuentra [24] en donde los autores realizan la programación de personal de salud en una institución médica utilizando las capacidades de pronóstico de los agentes que conforman éste sistema. En cuanto a aplicaciones multi-agente que traten específicamente el problema de cuidado domiciliario no fueron encontradas por el autor en la literatura disponible sobre el tema. Para mayor profundidad en el tema se sugiere revisar [101], [102] en donde se realizan revisiones de diferentes modelos y aplicaciones de sistemas multi-agente en los últimos años.

3.3. CONCLUSIONES DEL CAPÍTULO

El cuidado a la salud domiciliaria es una alternativa a la prestación convencional del servicio de salud la cual mejora las condiciones y la calidad de vida de los pacientes reduciendo los costos de la prestación del servicio de salud. El evidenciado desarrollo del sector del cuidado domiciliar responde a distintos factores como lo son el incremento y el envejecimiento de la población, avances tecnológicos que permiten una mejor comunicación con los pacientes, el aumento de pacientes con enfermedades crónicas y presiones para reducir costos en materia de salud. Distintos métodos y aproximaciones han sido utilizados para resolver problemas específicos en este sector puntualmente en el ámbito operativo del servicio.

Se concluye que la mayoría de trabajos existentes sobre el cuidado domiciliar abordan ambos problemas, el de asignación y de ruteo. Lo anterior se realiza de manera simultánea ya sea jerárquicamente (primero resolviendo el problema de asignación y después el ruteo) o incluyendo ambos componentes a la vez en los modelos matemáticos utilizados para tratar el problema. La minimización de los costos operacionales, de la distancia recorrida y del tiempo de espera representan los objetivos más comunes en las investigaciones realizadas. Las restricciones más consideradas son los horarios de visita a los pacientes (ventanas de tiempo), la habilidad del operador de salud que realizará la visita domiciliaria y los tiempos de trabajo de los operadores de salud en el sistema. Los métodos de solución más utilizados son las aproximadas a causa de que los problemas de asignación y de ruteo son de naturaleza NP-Completo y mediante un método aproximado se obtienen soluciones de buena calidad en un tiempo de computación corto. Son muy pocos los trabajos que utilizan métodos alternativos como lo son los sistemas expertos, la lógica difusa o los modelos basados en agentes. En algunos casos se cuenta con información real para probar las propuestas de los autores, en los otros casos las instancias encontradas en la literatura son utilizadas para realizar las pruebas, por lo que el origen de los datos de entrada no representa un problema a la hora de realizar las investigaciones. Por último se encontró que la mayoría de los trabajos no tienen en cuenta la incertidumbre ni el dinamismo del cuidado domiciliar por lo que desarrollan aproximaciones determinísticas y estáticas.

Durante el desarrollo de la revisión literaria del problema de asignación y ruteo se identificaron algunos vacíos que pueden representar algunas oportunidades de investigación como: (i) desarrollar modelos estocásticos los cuales consideren la incertidumbre en las actividades del cuidado domiciliar, (ii) construcción de novedosos procedimientos heurísticos y meta heurísticos con el fin de encontrar soluciones de buena calidad en un corto tiempo, (iii) utilización de otras aproximaciones como la lógica difusa, los sistemas expertos o los **modelos basados en agentes** explotando las ventajas de cada método, (iv) construir modelos de programación que permitan modificar en tiempo real los esquemas de asignación y ruteo en caso de imprevistos teniendo en cuenta la característica dinámica del cuidado domiciliar, (v) desarrollar herramientas computacionales que integren modelos de optimización con bases de datos, mapas, rutas o medios de transporte, con el fin de que las investigaciones en el cuidado domiciliar tengan incidencia en las organizaciones que proveen el servicio y en la industria del cuidado domiciliar.

Por otra parte y respecto a la conceptualización en simulación basada en agentes se concluye que para desarrollar el sistema multi-agente del problema de ruteo de cuidado domiciliario, es necesario desarrollar etapas de conceptualización, análisis y diseño, en donde se definan los agentes, las interacciones y el objetivo general del sistema. Finalmente de ésta sección se concluye que se utilizará la librería JADE de Java para la implementación del sistema multi-agente debido a que JADE hace énfasis en la comunicación facilitando el envío y recepción de mensajes entre los agentes o entidades que hacen parte del sistema, permite definir claramente los comportamientos y métodos de los agentes inteligentes y además es un software de distribución libre.

A manera de conclusión general y con base en los hallazgos de la revisión del estado del arte del problema de ruteo de operadores de salud y de la conceptualización sobre simulación basada en agentes se realizará lo siguiente:

- Un modelo de programación matemática en el cual el costo de la prestación del servicio y la satisfacción de los pacientes sean los objetivos.
- Las principales restricciones que manejará el modelo de programación matemática estarán relacionadas con las habilidades de los operadores de salud, la salida y el retorno a los hogares de los operadores de salud después de atender a los pacientes (multi-depósito), el tiempo de trabajo de los operadores de salud, entre otras.
- Se utilizará el software XPRESS para la implementación y experimentación del modelo de programación matemática, debido a la facilidad que brinda a la hora de construir el código computacional del modelo y debido a la licencia estudiantil con que cuenta la universidad la cual permite resolver problemas de gran tamaño sin limitar el número de variables ni de restricciones.
- Se construirá un sistema multi-agente utilizando las metodologías mostradas en la etapa de conceptualización de sistemas multi-agente, en donde a través de las etapas de conceptualización, análisis y diseño en donde utilizando diagramas de casos de uso, de secuencia, de roles, entre otros se definirán las entidades (agentes), los métodos de operación y las interacciones del sistema de cuidado a la salud domiciliaria.
- El sistema multi-agente se implementará en Java utilizando la librería de agentes JADE en el entorno de desarrollo integrado (IDE por sus siglas en inglés) Eclipse.
- Se elaborarán simulaciones preliminares para validar la implementación del sistema multi-agente.
- Se realizará un estudio de caso que es construido partiendo de información real y agregando unos supuestos con el fin de validar el sistema multi-agente así como el trabajo realizado durante el proyecto de investigación.

3.4. RESULTADOS DE INVESTIGACIÓN

El contenido de este capítulo, específicamente del numeral 3.1. ha sido sometido para revisión a la revista ESTUDIOS GERENCIALES: López-Santana Eduyn R., Espejo-Díaz Julián A., Méndez-Giraldo German (2016) "Asignación y ruteo en el cuidado de la salud domiciliaria: revisión de la literatura". *Estudios gerenciales*.

Capítulo 4

MODELO MATEMÁTICO PARA EL RUTEO DE OPERADORES DE SALUD EN EL CUIDADO DOMICILIARIO

4.1. DEFINICIÓN DEL PROBLEMA

Se parte de un sistema de cuidado a la salud domiciliaria en el cual se considera un conjunto de pacientes $P = \{1, 2, \dots, N\}$ los cuales están distribuidos geográficamente sobre una región. El proveedor del servicio cuenta con un conjunto de operadores de salud $K = \{1, 2, \dots, M\}$ los cuales son heterogéneos respecto al nivel de preparación y habilidades. Cada operador de salud inicia y finaliza su recorrido en su respectivo hogar, además que cuenta con un tiempo máximo de trabajo determinado.

Dado lo anterior el problema es definido mediante un grafo no dirigido $G = (V, A)$ con un conjunto de nodos $V = KI \cup P \cup KF$, donde KI representa los nodos de inicio de los operadores de salud, P los hogares de los pacientes y KF los nodos ficticios que representan el destino de los operadores de salud donde $KI = KF$. El conjunto de arcos es definido por $A = \{(i, j) : i, j \in V, i \neq j\}$ y cada arco (i, j) tiene asociado un valor no negativo t_{ij} que representa el tiempo de viaje desde una ubicación i a otra ubicación j . Cada paciente $i \in P$ presenta un diagnóstico médico que requiere de un nivel de habilidad mínimo h_i siendo 1 el más bajo. La prioridad en la atención es determinada por el parámetro p_i siendo 1 el más alto e indica que el paciente debe ser visitado en la mayor brevedad posible. La visita domiciliaria al paciente tiene una duración determinada s_i y debe comenzar después del momento a_i en que comunicó la necesidad. Por otra parte a cada operador de salud $k \in K$ se le establece un nivel de habilidad o experticia q_k . El anterior parámetro es utilizado para lograr que la visita domiciliaria sea realizada sólo si el nivel de calificación del operador de salud q_k es mayor o igual al nivel de habilidad o calificación requerido por los pacientes h_i .

El valor mínimo que la promesa de servicio Z_i puede tomar para cada paciente i está calculado con base en los valores del TRIAGE correspondientes a [11] los cuales se encuentran en la Tabla 8. Para lograr lo anterior se construyó una función cuadrática de los tiempos recomendados de atención según su nivel de prioridad p_i como se observa en la Figura 5. De esa manera los coeficientes de la función cuadrática que serán utilizados en el modelo matemático son $\gamma = 16.071$, $\delta = -37.929$ y $\varepsilon = 24$.

Tabla 8 Tiempo de atención con base en el nivel de TRIAGE
Fuente: Tomado de [11]

Nivel TRIAGE	Tiempo de espera para atención (minutos)
1	0
2	15
3	60
4	120
5	240

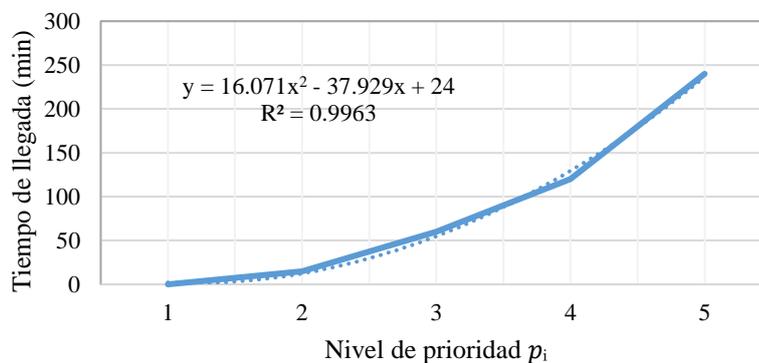


Figura 5 Función cuadrática del TRIAGE
Fuente: Autor

Es importante resaltar que los objetivos del modelo matemático son la minimización de los tiempos de viaje de los operadores de salud y la minimización de la promesa de servicio, es decir, que los operadores de salud lleguen a atender al paciente respetando los tiempos del TRIAGE en lo posible, como se observa en la Figura 6 en donde los tiempos de arribo a los pacientes deben seguir la dirección de las líneas hasta la cota mínima que es representada por la función de TRIAGE.

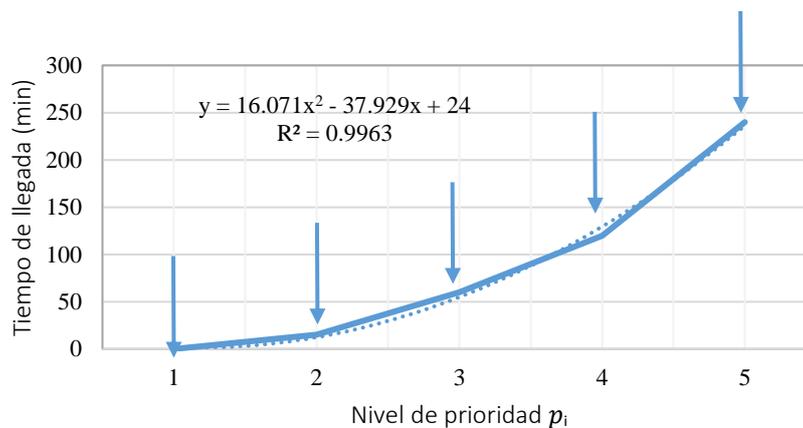


Figura 6 Minimización de la promesa de servicio
Fuente: Autor

Con base en el contenido de la definición del problema se elabora la Figura 7 la cual resume el problema de ruteo de cuidado domiciliario.

Cuidado a la salud domiciliaria

FO. 1 Minimizar tiempo de transporte
FO. 2 Minimizar promesa de servicio

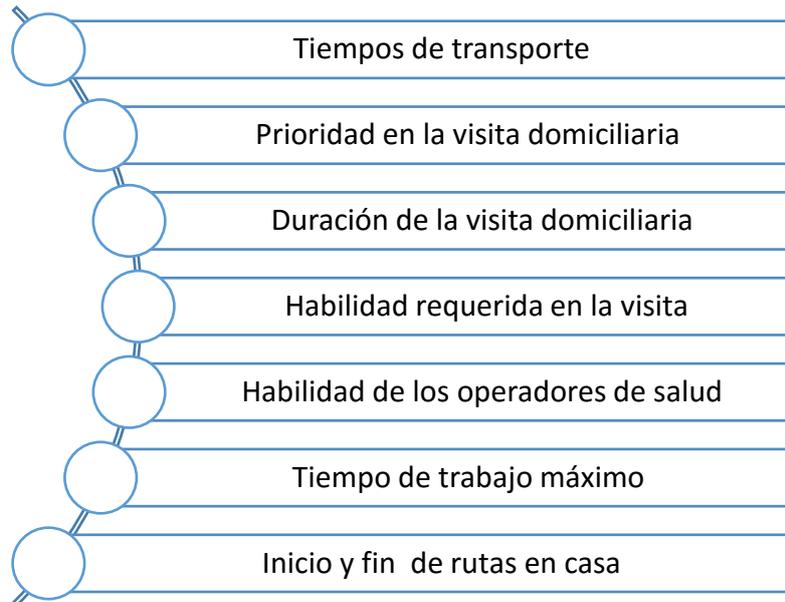


Figura 7 Problema de ruteo de cuidado domiciliario
Fuente: Autor

4.2. DESARROLLO DEL MODELO MATEMÁTICO

El modelo matemático será desarrollado mediante la programación entera mixta. A continuación se presenta la notación usada en el modelo matemático así como la formulación del modelo mediante las ecuaciones que serán explicadas posteriormente.

4.2.1. Notación del problema

La notación empleada para el problema y el modelo de programación entera es presentada en la Tabla 9.

Tabla 9 Notación usada para el modelo de programación entera mixta

Fuente: Autor

<p>Conjuntos</p> <p>P : Conjunto de pacientes</p> <p>K : Conjunto de operadores de salud</p> <p>KI : Conjunto de orígenes de los operadores de salud</p> <p>KF : Conjunto de destinos de los operadores de salud</p> <p>V : Conjunto de vértices ($V = KI + P + KF$)</p> <p>A : Conjunto de arcos</p> <p>$V1$: Conjunto de vértices 1 ($V1 = KI + P$)</p> <p>$V2$: Conjunto de vértices 2 ($V2 = P + KF$)</p>
<p>Parámetros</p> <p>t_{ij} : Tiempo de viaje desde el paciente i al paciente j.</p> <p>a_i : Momento en el cual el paciente i comunica la necesidad del servicio.</p> <p>s_i : Duración de la visita domiciliaria al paciente i.</p> <p>h_i : Habilidad mínima requerida por el paciente i . [1 Nivel de requerimiento más bajo].</p> <p>p_i : Prioridad en la atención del paciente i en el sistema. [1 Prioridad máxima].</p> <p>q_k : Habilidad del operador de salud k . [1 Nivel de experticia más bajo].</p> <p>l_k : Comienzo de la jornada laboral del operador de salud k.</p> <p>u_k : Finalización de la jornada laboral del operador de salud k.</p> <p>BM : Valor numérico alto para restricciones lógicas.</p> <p>α : Ponderación para la primera función objetivo.</p> <p>β : Ponderación para la segunda función objetivo.</p> <p>G : Tamaño del conjunto P más el tamaño del conjunto KI.</p> <p>$\gamma, \delta, \varepsilon$: Coeficientes de la función TRIAGE</p>
<p>Variables</p> <p>X_{ijk} : 1 Si el operador de salud k cubre la ruta $(i - j)$, 0 en otro caso.</p> <p>Y_{ik} : Momento de llegada del operador de salud k al paciente i</p> <p>Z_i : Promesa de servicio al paciente i</p>

4.2.2. Formulación del problema

El modelo matemático correspondiente a la programación y ruteo de operadores de salud es presentado en las ecuaciones (1) a la (21).

$$\text{Minimizar } FO = \alpha FO_1 + \beta FO_2 \quad (1)$$

$$FO_1 = \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} X_{ijk} t_{ij} \quad (2)$$

$$FO_2 = \sum_{i \in P} \frac{Z_i}{|P|} \quad (3)$$

Sujeto a.,

$$\sum_{k \in K} \sum_{i \in V1} X_{ijk} = 1 \quad \forall j \in P \quad (4)$$

$$\sum_{j \in V1} X_{jek} = \sum_{j \in V2} X_{ejk} \quad \forall k \in K, e \in P \quad (5)$$

$$\sum_{j \in V2} X_{ijk | i=k} = 1 \quad \forall k \in K, i \in KI \quad (6)$$

$$\sum_{i \in KI} \sum_{j \in V2} X_{ijk} = 0 \quad \forall k \in K \quad (7)$$

$$\sum_{i \in V1} X_{i \ k+G \ k} = 1 \quad \forall k \in K \quad (8)$$

$$\sum_{i \in V1} \sum_{j \in KF | j \neq k+G} X_{ijk} = 0 \quad \forall k \in K \quad (9)$$

$$Y_{ik} + t_{ij} + s_i - M(1 - X_{ijk}) \leq Y_{jk} \quad \forall k \in K, (i, j) \in P \quad (10)$$

$$Y_{ik} + t_{ij} - M(1 - X_{ijk}) \leq Y_{jk} \quad \forall k \in K, i \in KI, j \in V2 \quad (11)$$

$$Y_{ik} + t_{ij} - M(1 - X_{ijk}) \leq Y_{jk} \quad \forall k \in K, i \in V1, j \in KF \quad (12)$$

$$Y_{ik} \geq a_i \quad \forall k \in K, i \in P \quad (13)$$

$$\gamma p_i^2 - \delta p_i + \varepsilon + a_i \leq Z_i \quad \forall i \in P \quad (14)$$

$$Y_{ik} \leq Z_i \quad \forall k \in K, i \in P \quad (15)$$

$$h_j - M(1 - X_{ijk}) \leq q_k \quad \forall k \in K, i \in V1, j \in P \quad (16)$$

$$Y_{ik} \geq l_i \quad \forall k \in K, i \in KI \quad (17)$$

$$Y_{jk} \leq u_j \quad \forall k \in K, j \in KF \quad (18)$$

$$X_{ijk} \in \{0,1\} \quad \forall (i,j) \in V | i \neq j, k \in K \quad (19)$$

$$Y_{ik} \geq 0 \quad \forall i \in V, k \in K \quad (20)$$

$$Z_i \geq 0 \quad i \in P \quad (21)$$

La ecuación (1) representa la función multi-objetivo del sistema la cual está compuesta por la ecuación (2) que busca minimizar el tiempo de desplazamiento de operadores de salud y la ecuación (3) que busca minimizar demora en la prestación del servicio. La restricción (4) asegura que todos los pacientes sean atendidos. En (5) se asegura la conservación del flujo, es decir si un operador de salud visita un paciente éste debe irse. Las restricciones (6), (7), (8) y (9) garantizan que cada operador de salud salga y regrese a su propio hogar. Para los operadores de salud el tiempo de llegada a los pacientes, el tiempo de salida y regresos a sus hogares es calculado en (10), (11) y (12). La restricción (13) asegura que las visitas no sean realizadas antes de la notificación de la necesidad por parte del paciente. En (14) se calcula el mínimo de la promesa de servicio para cada paciente de acuerdo con su prioridad. En (15) se relaciona el tiempo de llegada del operador de salud con la promesa de servicio. La condición de que cada visita sea realizada por un operador de salud calificado es cubierta por (16). En (17) y (18) se observa la jornada laboral de los operadores de salud. En (19) se asegura que las variables X_{ijk} sean binarias y en (20) y (21) que el tiempo de llegada y la promesa de servicio sean positivas.

4.3. IMPLEMENTACIÓN DEL MODELO MATEMÁTICO

El modelo fue codificado en el software Xpress-Mp 7.9 en un computador personal con Windows 10, 6GB de RAM y un procesador de 2.5GHz. Se utilizó la licencia estudiantil adquirida por la Universidad Distrital Francisco José de Caldas a la cual se tuvo acceso a través del grupo de investigación Sistemas Expertos y Simulación (SES). Es importante resaltar que el uso de la licencia estudiantil permite ejecutar modelos de gran magnitud sin limitaciones de número de variables ni número de restricciones, sin la cual no es posible resolver el problema debido al gran número de variables y restricciones que tiene el modelo matemático presentado en el presente capítulo.

Durante la implementación del código computacional se observó la necesidad de calcular la matriz de tiempos de viaje entre ubicaciones t_{ij} en caso de que la información de entrada contenga las coordenadas de las ubicaciones de los pacientes, donde xx_i es la coordenada en X del paciente o del operador de salud y por otra parte yy_i es la coordenada en Y del paciente o del operador de salud. Para lograr lo anterior se calculan las distancias (rectangulares) entre las ubicaciones y se multiplica por un factor de velocidad FV obteniendo de esta manera los tiempos de transporte entre las ubicaciones. Para efectos de lo anterior se agregan los siguientes parámetros al modelo matemático de ser necesario además de las siguientes igualdades.

$$t_{ij} = (|xx_j - xx_i| + |yy_j - yy_i|) \times FV \quad \forall i \in KI, j \in P \quad (22)$$

$$t_{i\ j+|V|} = (|xx_j - xx_i| + |yy_j - yy_i|) \times FV \quad \forall i \in P, j \in KI \quad (23)$$

$$t_{ij} = (|xx_j - xx_i| + |yy_j - yy_i|) \times FV \quad \forall (i, j) \in P \mid i \neq j \quad (24)$$

$$t_{ij} = 100000 \times FV \quad \forall (i, j) \in P \mid i = j \quad (25)$$

En (22) se calcula la matriz de tiempos de desplazamiento entre los hogares de los operadores de salud y los hogares de los pacientes, en (23) entre los hogares de los pacientes y los destinos de los operadores de salud (sus hogares), la igualdad (24) muestra el cálculo de la matriz de tiempos de desplazamiento entre las ubicaciones de los pacientes y finalmente en (25) se penalizan los tiempos de desplazamiento entre las mismas ubicaciones para que no hagan parte de la solución obligando que los operadores de salud no se queden en la misma ubicación. Por otra parte en la implementación del modelo matemático se utilizaron archivos planos de texto (extensión *.txt) para manejar la información de entrada de los operadores de salud, la información de los pacientes y los coeficientes de la función del TRIAGE. Finalmente es importante resaltar que en el Anexo 2 se encuentra el código computacional construido en Xpress para el modelo de programación matemático propuesto.

4.4. EXPERIMENTACIÓN Y RESULTADOS

Con el fin de evaluar el desempeño del modelo de programación matemática propuesto se utilizaron las instancias diseñadas por Cordeau (pr01 y pr02) disponibles en [103]. Estas instancias corresponden al problema de ruteo de vehículos con ventanas de tiempo y múltiples depósitos a las cuales se agregó los parámetros de interés del modelo. Se resolvió el problema para 12 pacientes y 4 operadores de salud para cada instancia variando los parámetros α y β en cada caso. Es importante resaltar que en la información de la instancia se encuentra la localización o las coordenadas de los operadores de salud y de los pacientes y no los tiempos de transporte. Por lo tanto los tiempos t_{ij} entre las ubicaciones fueron calculados utilizando distancias rectangulares considerando un factor de velocidad de 1 añadiendo (22), (23), (24) y (25) al modelo en Xpress. En la Tabla 10 y la Tabla 11 se presenta la información de los operadores de salud y de los pacientes respectivamente para la instancia pr01.

Tabla 10 Información de operadores de salud (pr01)

Fuente: Adaptado de [103]

Operador de salud	Coordenada X	Coordenada Y	Habilidad (q_k)	Tiempo de inicio (l_k)	Tiempo finalización (u_k)
1	4.163	13.559	3	0	500
2	21.387	17.105	5	0	500
3	-36.118	49.097	3	0	500
4	-31.201	0.235	3	0	500

Tabla 11 Información de pacientes (pr01)

Fuente: Adaptado de [103]

Paciente	Coordenada X	Coordenada Y	Momento de notificación (a_i)	Duración del servicio (s_i)	Habilidad requerida (h_i)	Prioridad (p_i)
1	-29.73	64.136	399	2	5	2
2	-30.664	5.463	121	7	1	2
3	51.642	5.469	389	21	2	2
4	-13.171	69.336	204	24	5	3
5	-67.413	68.323	317	1	4	3
6	48.907	6.274	160	17	3	2
7	5.243	22.26	170	6	4	3
8	-65.002	77.234	215	5	5	2
9	-4.175	-1.569	80	7	4	2
10	23.029	11.639	90	1	3	3
11	25.482	6.287	397	15	2	4
12	-42.615	-26.932	271	25	2	4

MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA

En la Tabla 12 se presentan los resultados variando los parámetros α y β correspondientes a la ponderación de la función objetivo de tiempo de desplazamiento (FO_1) y de promesa de servicio (FO_2) respectivamente. En la Tabla 13 y la Tabla 14 se presentan los esquemas de ruteo para distintos valores de α y β .

Tabla 12 Resultados de la prueba variando α y β para instancia pr01

Fuente: Autor

α	β	FO	FO_1	FO_2
1	0	535.036	535.036	0
0.8	0.2	489.347	535.036	306.592
0.6	0.4	443.658	535.036	306.592
0.4	0.6	394.728	553.872	288.633
0.2	0.8	341.680	553.872	288.633
0	1	280.49	0	280.49

Tabla 13 Esquema de ruteo para $\alpha=0.2$ $\beta=0.8$ instancia pr01

Fuente: Autor

Operador de salud 1		Operador de salud 2		Operador de salud 3		Operador de salud 4	
DE	A	DE	A	DE	A	DE	A
H1	P10	H2	P9	H3	H3	H4	P2
P10	P6	P9	P7	-	-	P2	P12
P6	P3	P7	P4	-	-	P12	H4
P3	P11	P4	P8	-	-	-	-
P11	H1	P8	P5	-	-	-	-
-	-	P5	P1	-	-	-	-
-	-	P1	H2	-	-	-	-

Tabla 14 Esquema de ruteo para $\alpha=0.8$ $\beta=0.2$ instancia pr01

Fuente: Autor

Operador de salud 1		Operador de salud 2		Operador de salud 3		Operador de salud 4	
DE	A	DE	A	DE	A	DE	A
H1	P10	H2	P9	H3	H3	H4	P2
P10	P6	P9	P4	-	-	P2	P12
P6	P3	P4	P8	-	-	P12	H4
P3	P11	P8	P5	-	-	-	-
P11	H1	P5	P1	-	-	-	-
-	-	P1	P7	-	-	-	-
-	-	P7	H2	-	-	-	-

Según los resultados de ésta experimentación (instancia pr01) se puede observar como al incrementar α (o disminuir β) la función objetivo referente a los tiempos de viaje (FO_1) aumenta y disminuye la función objetivo de promesa de servicio (FO_2). Además se evidencia que el esquema de ruteo cambia para el operador de salud 2 cuando se incrementa α y se disminuye β .

MODELO MATEMÁTICO PARA EL RUTEO DE OPERADORES DE SALUD EN EL CUIDADO DOMICILIARIO

A continuación se realiza otra experimentación utilizando en ésta ocasión la instancia pr02 de Cordeau. En la Tabla 15 y la Tabla 16 se presenta la información de los operadores de salud y de los pacientes respectivamente para la instancia pr02. En la Tabla 17 se presentan los resultados de la prueba variando α y β obteniendo los esquemas de ruteo de Tabla 18 y la Tabla 19.

Tabla 15 Información de operadores de salud instancia pr02

Fuente: Adaptado de [103]

Operador de salud	Coordenada X	Coordenada Y	Habilidad (q_k)	Tiempo de inicio (l_k)	Tiempo finalización (u_k)
1	6.229	10.590	3	0	1000
2	32.663	44.730	5	0	1000
3	48.807	48.792	3	0	1000
4	33.179	-4.968	3	0	1000

Tabla 16 Información de pacientes instancia pr02

Fuente: Adaptado de [103]

Paciente	Coordenada X	Coordenada Y	Momento de notificación (a_i)	Duración del servicio (s_i)	Habilidad requerida (h_i)	Prioridad (p_i)
1	33.588	30.750	410	12	5	2
2	48.828	65.314	539	1	1	2
3	86.176	59.344	125	1	2	2
4	39.270	-33.057	278	15	5	3
5	-23.370	89.853	256	10	4	3
6	48.132	95.593	378	12	3	2
7	-16.357	93.311	382	15	4	3
8	-57.703	-65.601	504	16	5	2
9	7.147	32.648	422	4	4	2
10	42.950	68.701	539	3	3	3
11	37.085	-2.112	336	2	2	4
12	77.759	55.817	444	2	2	4

Tabla 17 Resultados de la prueba variando α y β para instancia pr02

Fuente: Autor

α	β	FO	FO_1	FO_2
1	0	715.988	75.988	0
0.8	0.2	698.361	728.73	576.887
0.6	0.4	662.204	742.254	542.13
0.4	0.6	622.179	742.254	542.13
0.2	0.8	560.512	824.51	494.512
0	1	430.845	0	430.485

MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA

Tabla 18 Esquema de ruteo para $\alpha=0.2$ $\beta=0.8$ instancia pr02

Fuente: Autor

Operador de salud 1		Operador de salud 2		Operador de salud 3		Operador de salud 4	
DE	A	DE	A	DE	A	DE	A
H1	H1	H2	P5	H3	P3	H4	P11
-	-	P5	P7	P3	P6	P11	H4
-	-	P7	P9	P6	P10	-	-
-	-	P9	P1	P10	P2	-	-
-	-	P1	P4	P2	P12	-	-
-	-	P4	P8	P12	H3	-	-
-	-	P8	H2	-	-	-	-

Tabla 19 Esquema de ruteo para $\alpha=0.8$ $\beta=0.2$ instancia pr02

Fuente: Autor

Operador de salud 1		Operador de salud 2		Operador de salud 3		Operador de salud 4	
DE	A	DE	A	DE	A	DE	A
H1	H1	H2	P11	H3	P6	H4	H4
-	-	P11	P4	P6	P10	-	-
-	-	P4	P8	P10	P2	-	-
-	-	P8	P5	P2	P3	-	-
-	-	P5	P7	P3	P12	-	-
-	-	P7	P9	P12	H3	-	-
-	-	P9	P1	-	-	-	-
-	-	P1	H2	-	-	-	-

Según los resultados de ésta experimentación (instancia pr02) y como en la experimentación inicial se puede observar como al incrementar α (o disminuir β) la función objetivo referente a los tiempos de viaje (FO1) aumenta y disminuye la función objetivo de promesa de servicio (FO2). Por otra parte se puede evidenciar que con un $\alpha=0.2$ y un $\beta=0.8$, los cuales indican que la promesa de servicio tiene más importancia que el tiempo de transporte se utilizan tres operadores de salud para atender los pacientes (OP1, OP2 y OP4); mientras que con un $\alpha=0.8$ y un $\beta=0.2$, los cuales indican que el tiempo de transporte tiene más importancia que la promesa de servicio se utilizan sólo dos operadores de salud para atender a los pacientes (OP2 y OP4).

Respecto al tiempo de ejecución del modelo, se observa que éste no superó los 100 segundos para ninguna ejecución del modelo, por lo que para éste número de pacientes el esfuerzo computacional no representa ningún inconveniente. Sin embargo al utilizar más pacientes y más operadores de salud el tamaño del problema crece exponencialmente al ser un problema de tipo Np-completo

4.5. CONCLUSIONES

En este capítulo se presentó el problema de programación y ruteo de operadores de salud considerando la promesa de servicio en el cuidado domiciliario. Se propuso un modelo de programación entera mixta que resuelve el problema teniendo en cuenta las habilidades de los operadores de salud y las prioridades en la atención de los pacientes. Se consideraron dos funciones objetivo que fueron la minimización del tiempo de desplazamiento y la minimización de la promesa de servicio. Se utilizó programación multi-objetivo utilizando ponderación para darle más valor a una función que otra. Mediante los ejemplos de aplicación se encontró que el esquema de ruteo varía si se modifican las ponderaciones de las funciones objetivo además de que también varía el número de operadores de salud utilizados para realizar los servicios domiciliarios.

Trabajos futuros apuntan a la construcción de procedimientos heurísticos o metaheurísticos que permitan resolver instancias de gran magnitud en un tiempo razonable. También se contempla la construcción de un modelo multi agente que permita resolver varias veces el modelo de programación matemática en distintos periodos de tiempo.

4.6. RESULTADOS DE INVESTIGACIÓN

El contenido de éste capítulo fue presentado y aceptado para presentación oral en el “III Congreso Internacional Industria y Organizaciones - CIIO 2016” a través del artículo: E. López-Santana, J. Espejo-Díaz, and G. Méndez-Giraldo, “Modelo de programación entera mixta para programación y ruteo en cuidado a la salud domiciliaria considerando la promesa de servicio,” in III Congreso Internacional de Industria y Organizaciones – “Gestión de Cadenas de Abastecimiento en un Mundo Cambiante,” 2016, p. in press, accepted.

Capítulo 5

APROXIMACIÓN MULTI-AGENTE DEL PROBLEMA DE RUTEO EN EL CUIDADO DOMICILIARIO

Un sistema de cuidado a la salud domiciliario puede ser visto como una red de servicios de salud en el cual participan los pacientes, la institución que presta el servicio, las personas involucradas en la implementación logística, las personas involucradas en los aspectos financieros y el equipo médico que se requiere para la prestación del servicio [8]. De la interacción de los anteriores actores o participantes emerge el comportamiento general del sistema el cual tiene el objetivo de prestar el servicio de cuidado a la salud en casa. Lo anterior nos sugiere que éste sistema puede ser modelado mediante el paradigma multi-agente debido a que existe una clara diferenciación de las entidades que componen el sistema, se pueden extraer las interacciones y el proceso de comunicación entre los participantes además de que las entidades actúan a favor de cumplir el objetivo general del sistema.

Con el fin de modelar el sistema de cuidado domiciliario como un sistema multi-agente se seguirá parte de las metodologías de Moreno y de Macal que se encuentran en la sección 3.2.1 del presente documento. Para lograr lo anterior se desarrollarán las etapas de conceptualización en donde se define claramente las entidades y su interacción, la etapa de análisis en donde se definen las actividades de cada agente y la comunicación entre ellos y finalmente la etapa de diseño en donde se especifica cómo se va a realizar lo dispuesto en las dos fases anteriores en la implementación computacional. Es importante resaltar que durante las etapas se mostrarán diagramas tipo UML para facilitar en la comprensión del sistema multi-agente. Por otra parte se aclara que el alcance del modelo multi-agente del sistema de cuidado domiciliario sólo involucra la parte operativa del sistema, por lo tanto no se incluyen los aspectos que no tienen relación a la planeación operativa como lo es el aspecto administrativo del sistema de cuidado domiciliario. Es por lo anterior que sólo se tendrán en cuenta las entidades que participan en el proceso de programación y ruteo de operadores de salud.

5.1. ETAPA DE CONCEPTUALIZACIÓN DEL SISTEMA

En ésta etapa como se indicó anteriormente se definirán puntualmente las entidades o los agentes que comprende el sistema multi-agente de cuidado domiciliario. Posteriormente mediante un diagrama de casos de uso se definirán las interacciones entre los agentes que previamente se definieron. Finalmente utilizando una plantilla de casos de uso se dará explicación al diagrama de casos de uso presentado.

5.1.1. Definición de las entidades

Las entidades que hacen parte de la programación y ruteo de operadores de salud son los pacientes, el organizador del sistema, el coordinador del sistema y finalmente los operadores de salud que visitan a los pacientes. A continuación se realiza una definición de las anteriores entidades y se hace una breve descripción de las actividades que realizan en el sistema de cuidado domiciliario.

a) Paciente

Entidad que presenta complicaciones que alteran su estado habitual de salud, para lo cual requiere de una atención médica lo suficientemente calificada y personalizada la cual debe ser llevada en su hogar. Para lograr recibir la atención domiciliaria calificada el paciente debe contactarse con el proveedor del servicio de cuidado domiciliario y debe describir sus síntomas y dar la ubicación en que se encuentra para recibir el servicio. De ésta manera el paciente busca evitar desplazamientos innecesarios a centros de salud y largas esperas de una cita médica.

b) Organizador

Entidad contratada por el proveedor del servicio de cuidado domiciliario la cual se encarga de registrar y parametrizar las solicitudes de los pacientes. Para lograr lo anterior debe construir un diagnóstico médico con base en los síntomas de los pacientes con el fin de establecer el nivel de prioridad en la atención, el nivel de especialidad requerida la visita domiciliaria y la duración del servicio. Es importante resaltar que el organizador debe comunicar la anterior información en forma de parámetros los cuales serán listados a continuación.

- Momento en el cual el paciente comunica la necesidad del servicio (a_i).
- Duración de la visita domiciliaria al paciente (s_i).
- Habilidad mínima requerida por el paciente (h_i) [1 Nivel de requerimiento más bajo].
- Nivel de prioridad en la atención del paciente en el sistema (p_i) [1 Prioridad máxima].
- Ubicación de los pacientes (dirección o coordenadas) (xx_i, yy_i).

Por otra parte el organizador debe conocer el nivel de habilidades de los operadores de salud que estén actualmente activos en el sistema, la anterior información también debe ser comunicada como parámetro al agente coordinador.

Finalmente el organizador debe conocer las ubicaciones de los operadores de salud, es decir, si se encuentran en sus hogares o se encuentran en la ubicación de algún paciente, de la misma forma ésta información debe ser comunicada como un parámetro al coordinador.

c) Coordinador

Entidad contratada por el proveedor del servicio de cuidado domiciliario la cual se encarga de realizar los esquemas de ruteo de los operadores de salud con base en la información recibida por parte del agente coordinador. Para lograr lo anterior el coordinador utiliza el modelo matemático presentado en la sección 4.2 del presente documento. Por último el coordinador informa a cada operador de salud su esquema de ruteo.

En otras palabras el coordinador es el que decide (con base en el modelo matemático) y comunica la secuencia de pacientes a los cuales los operadores de salud realizarán las atenciones médicas domiciliarias. Además de lo anterior el coordinador debe informarles a los operadores de salud el tiempo de llegada a los hogares de cada paciente. En caso de que el coordinador considere que el operador de salud no saldrá de su casa no le enviará ningún esquema de ruteo.

d) Operador de salud

Entidad especializada en atención médica la cual es contratada por el proveedor del servicio de cuidado domiciliario que encarga de realizar las visitas domiciliarias siguiendo el esquema de ruteo recibido por parte del coordinador. Debe comunicar al organizador los resultados de las visitas domiciliarias así como su ubicación y el tiempo desde el cual está disponible para recibir nuevos pacientes.

5.1.2. Definición de las interacciones

En éste apartado de capítulo se profundizará sobre las interacciones de las entidades del sistema de cuidado domiciliario. Para realizar lo anterior se utilizará un diagrama de casos de uso que el cual se muestra en la Figura 8, además de la correspondiente plantilla de casos de uso que se encuentra en la Tabla 20.

Es importante resaltar que un diagrama de casos de uso según los autores Rumbaugh, Jacobson y Booch [104] captura el comportamiento de un sistema o de un subsistema tal como se muestra a un usuario exterior. Por otra parte los autores indican que éste diagrama reparte la funcionalidad del sistema en transacciones para los actores del sistema, en donde las piezas de funcionalidad se llaman casos de uso. Adicionalmente se indica que el diagrama utiliza la notación <<exclude>> que se refiere a que un caso extiende el comportamiento del caso de uso base, y la notación <<include>> para demostrar dependencia del caso de uso base. Finalmente se presenta la plantilla del diagrama para mayor entendimiento del diagrama de casos de uso.

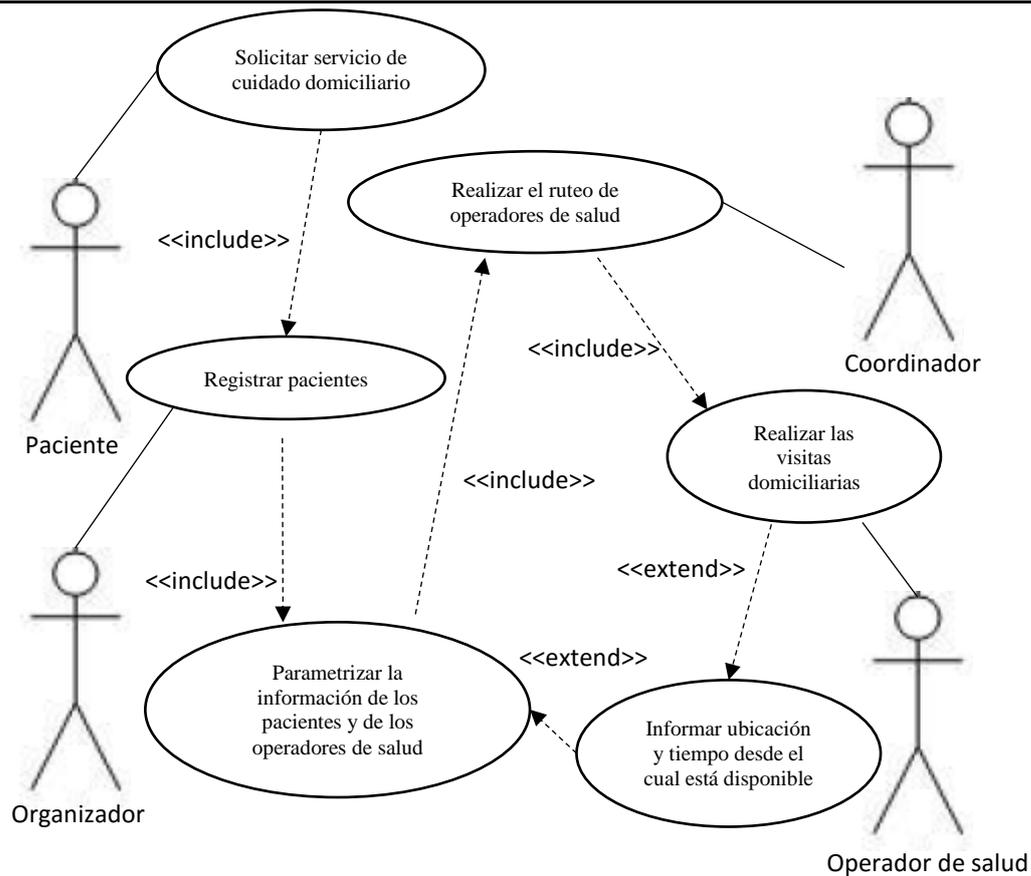


Figura 8 Diagrama de casos de uso del problema de cuidado domiciliario

Fuente: Autor

Tabla 20 Plantilla de caso de uso del problema de cuidado domiciliario

Fuente: Autor

Actores	Pacientes, Organizador, Coordinador y Operadores de salud.
Objetivo	Brindar el servicio de cuidado domiciliario mediante el ruteo de operadores de salud a los hogares de los pacientes que requieran el servicio.
Descripción	El paciente solicita el servicio de cuidado domiciliario, su solicitud es registrada y su información es parametrizada. Con la información anterior se realiza el ruteo utilizando un modelo matemático el cual es distribuido a los operadores de salud los cuales realizarán las visitas domiciliarias.
Pre-condición	Debe existir al menos una solicitud del servicio de cuidado domiciliario para que el sistema funcione
Flujo Alternativo	Si no hay una solicitud de un paciente no se realiza el ruteo de operadores de salud
Pos-condición	Después de que se realizó una visita domiciliaria debe haber una notificación por parte del operador de salud que la realizó.

5.2. ETAPA DE ANÁLISIS DEL SISTEMA

En ésta etapa mediante la utilización de distintos diagramas tipo UML se representan las actividades desarrolladas por cada entidad y la comunicación entre tales entidades.

5.2.1. Actividades

En la Figura 9 se presenta el diagrama de actividades del sistema en el cual se detallan las actividades que debe realizar cada entidad en el sistema.

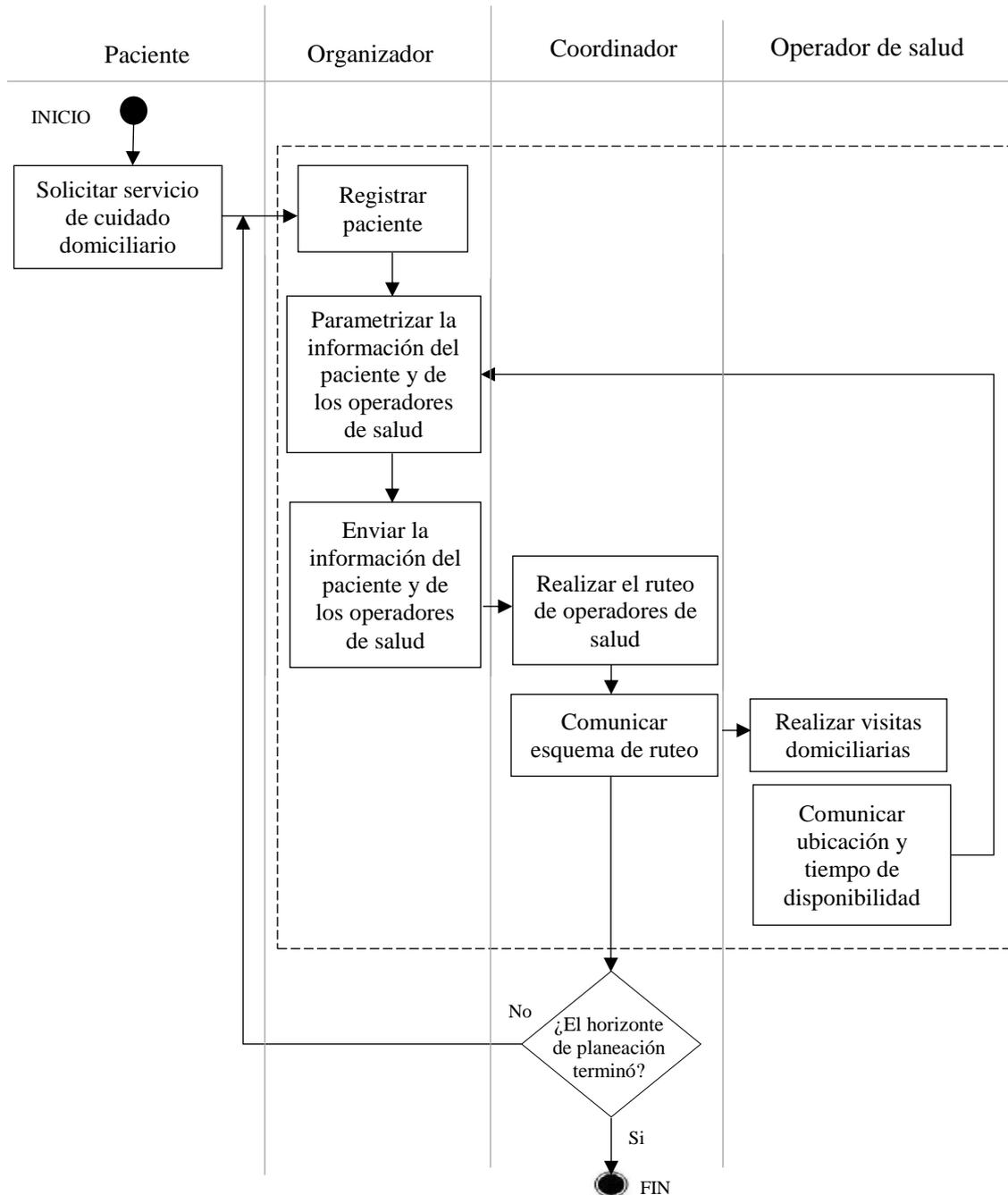


Figura 9 Diagrama de actividades del sistema de cuidado domiciliario

Fuente: Autor

El anterior diagrama presenta las actividades de cada entidad del sistema en la forma de flujo, el cual comienza cuando se genera una solicitud del servicio de cuidado domiciliario y finaliza cuando se termina el horizonte de planeación. Es importante resaltar que lo que se encuentra al interior de las líneas punteadas es realizado cada vez que se recoge cierta cantidad de solicitudes, es decir el ruteo se realiza una vez para cada grupo de solicitudes así como las demás actividades relacionadas a ésta actividad. Por otra parte en el mismo recuadro punteado se ejecutan las actividades de comunicación en el sistema entre el organizador, coordinador y los operadores de salud.

5.2.2. Comunicación e interacción

Con el fin detallar la comunicación entre las diferentes entidades del sistema se presenta la Figura 10 en donde se encuentra el diagrama de secuencia en el sistema. En el diagrama se muestra la interacción entre las entidades a través de tiempo, también se muestra los mensajes que serán enviados y recibidos durante la ejecución del modelo de simulación basado en agentes. Es importante resaltar que éste diagrama se muestra para un solo periodo de tiempo, por lo tanto la secuencia mostrada en la ilustración se repite para todo el horizonte de planeación el número de periodos del horizonte de planeación.

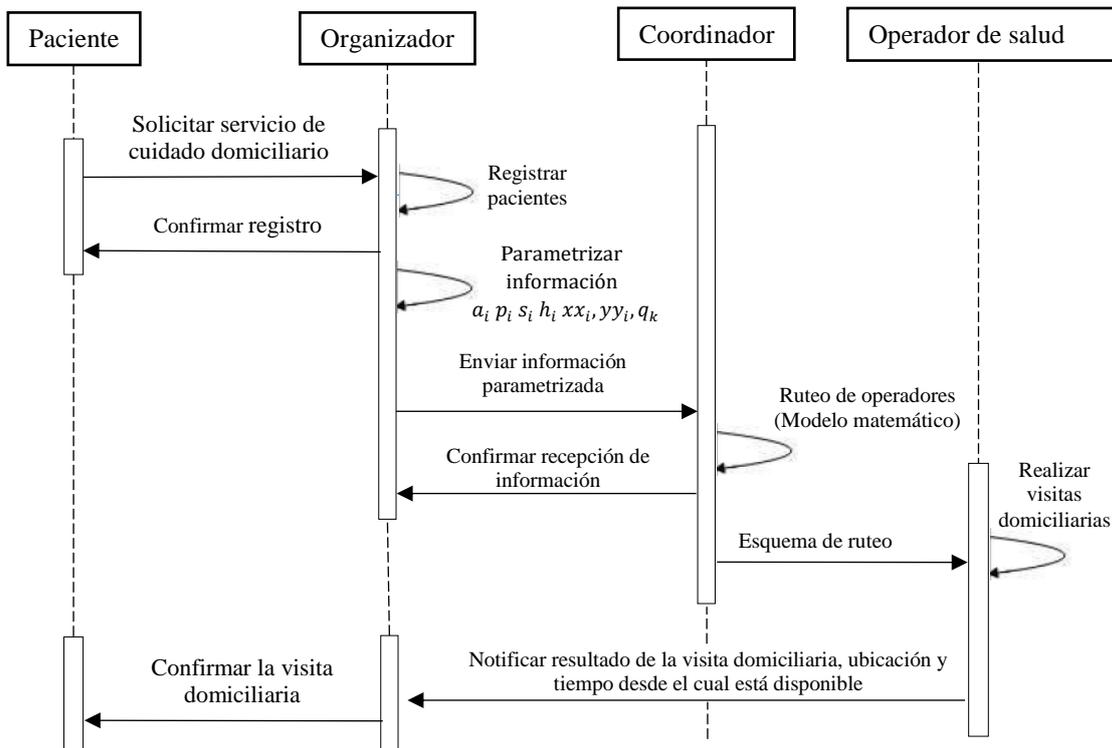


Figura 10 Diagrama de secuencia del sistema de cuidado domiciliario
Fuente: Autor

5.3. ETAPA DE DISEÑO DEL SISTEMA

En ésta etapa se establecerá la manera en que las entidades realizarán lo mostrado en las secciones anteriores. Por lo tanto se muestra un modelo de agentes o árbol de agentes y luego se especifican las reglas generales del sistema. Lo anterior se realiza con el fin de aclarar los conceptos y lograr un mayor entendimiento al momento de realizar la implementación en la plataforma para modelamiento y simulación de sistemas multi-agente.

5.3.1. Modelo de agentes

Como se indicó anteriormente en la etapa de diseño se alistará la implementación del sistema, por lo tanto se debe definir los roles de los agentes computacionales a construir. La entidad paciente y la entidad organizador se unen formando el agente generador el cual creará las solicitudes del sistema con base en su conocimiento del sistema y con base en información histórica para luego distribuir la información anterior a las partes interesadas. Por otra parte las funciones del coordinador serán realizadas por el agente coordinador de igual forma para la entidad operador de salud que se convierte en o los agentes operadores de salud. En la

Figura 11 se observa lo anteriormente descrito mediante un diagrama de roles también llamado árbol de agente.

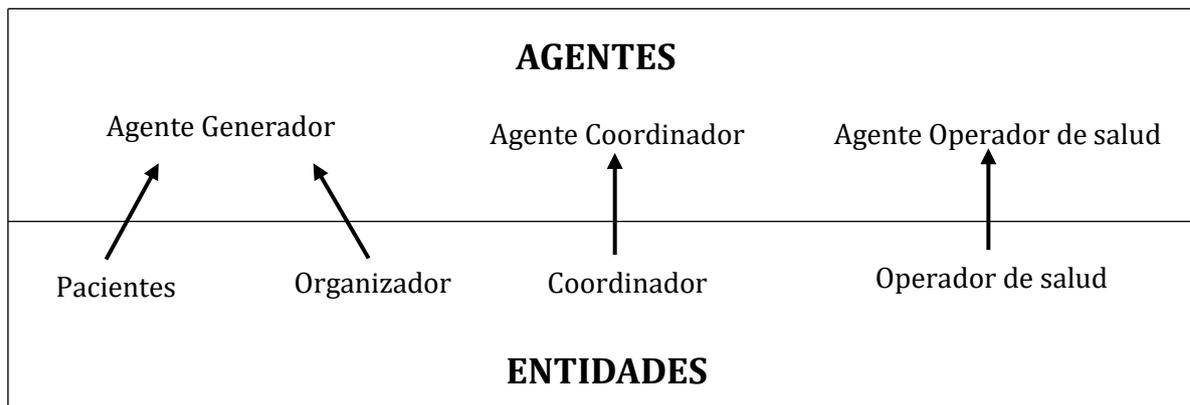


Figura 11 Diagrama de árbol de agentes
Fuente: Autor

5.3.2. Reglas del sistema basado en agentes

A continuación se especifican las reglas que regirán el comportamiento general del sistema las cuales no han sido especificadas anteriormente.

- El sistema trabaja por periodos en los cuales se reciben solicitudes que son atendidas o servidas en el siguiente periodo, es decir se acumulan solicitudes hasta cierto momento en el cual se comunica la información de los pacientes del sistema al coordinador que realiza el esquema de ruteo, los operadores de salud estarán disponibles hasta el siguiente periodo o hasta después del comienzo del siguiente periodo si el esquema de ruteo recibido así lo determina.
- Los operadores de salud comienzan la jornada laboral en sus hogares desde donde se dirigen a los pacientes los cuales les ha sido asignados en el periodo específico, se quedan en la ubicación del último paciente de la ruta hasta que reciban nuevos pacientes. En caso de que no reciban ningún esquema de ruteo no salen de sus hogares.

Por otra parte en la Figura 12 se muestra el esquema de la manera en que se ejecutará la aproximación multi-agente, la cual es mediante periodos en los cuales se reciben las solicitudes que serán atendidas en el siguiente periodo y paralelamente se atienden las solicitudes generadas el periodo anterior. Lo anterior es realizado en el horizonte de planeación del sistema.

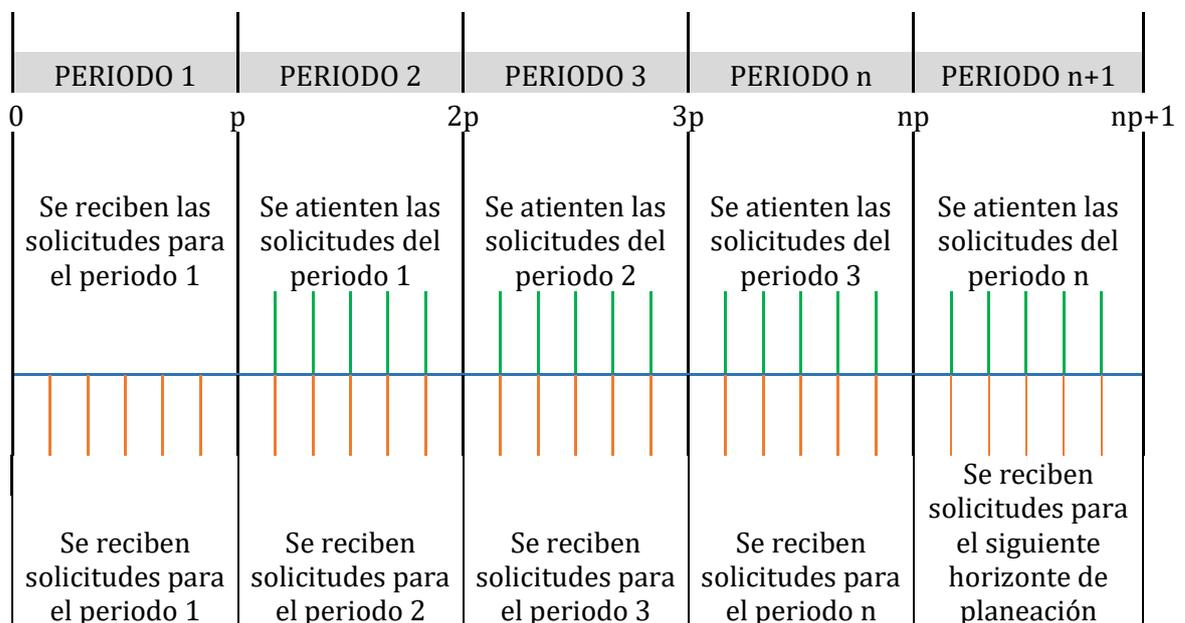


Figura 12 Configuración de la aproximación multi-agente
Fuente: Autor

5.4. CONCLUSIONES DEL CAPÍTULO

En el presente capítulo se presentó el sistema multi-agente del cuidado a la salud domiciliaria. Para ello se definió cada entidad y se caracterizaron los comportamientos, actividades e interacciones en el sistema. Lo anterior fue realizado mediante diagramas de caso de uso, de actividades, de secuencia y de roles. Por otra parte se especificaron las reglas generales que regirán la implementación computacional del sistema multi-agente de cuidado domiciliar. Se concluye que la mejor forma de manejar las solicitudes es por periodos, en donde en un periodo se recogen las solicitudes las cuales son atendidas en el siguiente periodo. Finalmente se realiza el supuesto de que los operadores de salud que se dirijan a realizar servicios domiciliarios deben quedarse en la ubicación del último paciente que visitaron en el periodo, deben informar al organizador su ubicación y con esa información se construye el esquema de ruteo para el siguiente periodo, de ésta forma ellos salen desde la última ubicación al hogar del primer paciente a atender en el siguiente periodo.

Capítulo 6

IMPLEMENTACIÓN DEL SISTEMA MULTI-AGENTE DE CUIDADO DOMICILIARIO

En este capítulo se presenta el desarrollo computacional del sistema multi-agente de cuidado domiciliario presentado en el capítulo anterior. Para ello se realiza una breve descripción del software y de las librerías utilizadas para construir el código computacional y se muestran los diagramas de clase del modelo de simulación basado en agentes. Por otra parte se muestra la secuencia de ejecución del código computacional del modelo con el fin de que el lector comprenda la manera en que opera el modelo de simulación basado en agentes. Finalmente se realizan pruebas del modelo variando los parámetros de entrada y analizando los resultados obtenidos de las pruebas de ejecución.

6.1. SOFTWARE EMPLEADO

6.1.1. Java y Eclipse

Para realizar la implementación del sistema multi-agente se utilizó el lenguaje de programación de propósito general **Java** en su versión Standard Edition 8 (JSE-1.8). Por otra parte al utilizar Java se puede emplear un entorno de desarrollo integrado (IDE por sus siglas en inglés) con el fin de editar el código fuente, compilar el código computacional, funcionar como depurador, entre otros. Para ello se utilizó el IDE **Eclipse** en su versión Neón.

6.1.2. Librerías

En el desarrollo del código computacional fue necesario utilizar librerías que corresponden a un conjunto de clases las cuales poseen una serie de métodos y atributos que facilitan la construcción del código además de que permiten el manejo del código de forma modular. A continuación se nombran y describen brevemente las librerías utilizadas.

JADE

Corresponde a la librería especializada en la construcción de sistemas multi-agente. Jade permite la creación de los agentes en diferentes hilos de ejecución (threads), por lo que en la configuración de la ejecución no se especifica un método principal (main) sino que se deben configurar los agentes del sistema en forma de argumentos como se observa en la Figura 13.

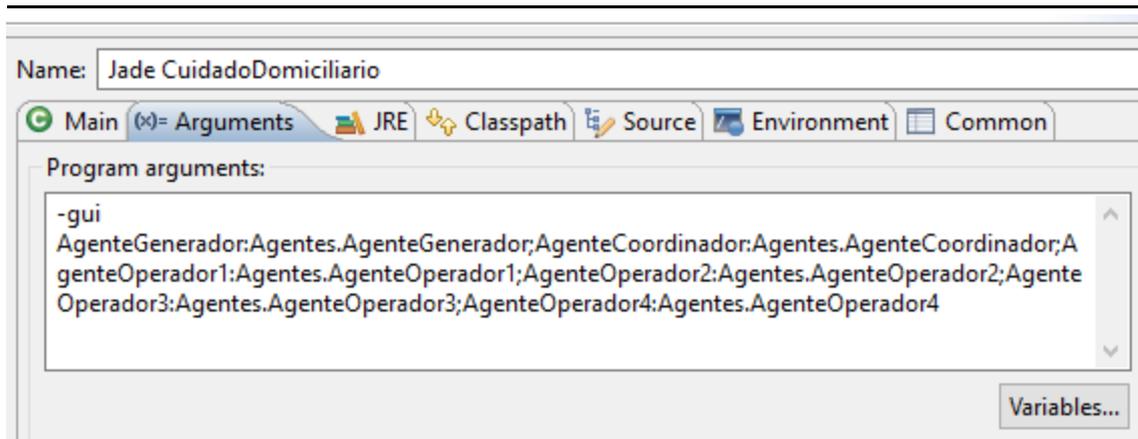


Figura 13 Configuración de Jade en Eclipse

Fuente: Autor

Por otra parte Jade permite establecer los comportamientos y métodos de los agentes además de que maneja los protocolos de comunicación en el sistema multi-agente, entre otros, mediante el importe de las distintas clases con que cuenta.

Librería de XRESS-MP (BLC)

Para acceder al software Xpress-MP desde Java se utilizó BLC (Builder Component Library), que corresponde a la interfaz de comunicación entre Xpress-MP y diversos lenguajes de programación, permite crear y ejecutar modelos directamente sin la intervención de programas externos. La idea de la utilización de BLC es tratar al optimizador de Xpress-MP como una librería por lo que mediante el importe de la librería *com.dashoptimization* a las clases que lo requieran se cuenta con el optimizador. BLC utiliza los siguientes objetos o clases para implementar el optimizador:

XPRB: Clase de conexión e inicialización

XPRBprob: Clase que maneja la definición del modelo

XPRBvar: Variable de decisión

XPRBexpr: Clase que maneja una expresión lineal

XPRBctr: Restricción

Java IO

Librería que contiene clases que permiten el flujo de datos en Java. Fue utilizada en la implementación multi-agente para crear, escribir, leer, almacenar en arreglos y sobrescribir archivos planos de texto.

Java Util

Librería que contiene las clases relacionadas con colecciones (listas, conjuntos, arreglos) permitiendo organizar los datos de una mejor forma. Fue utilizada en la implementación multi-agente para la creación y manejo de arreglos dinámicos, entre otros.

JSC (Java Statistical Classes)

Librería especializada en estadística la cual fue utilizada para la creación de variables que siguen una función de probabilidad determinada, manejar la creación de variables y números aleatorios facilitando el manejo de la probabilidad en el sistema multi-agente.

6.2. DIAGRAMAS DE CLASE DEL MODELO BASADO EN AGENTES

En este apartado de capítulo se muestran los diagramas de clase que exponen la construcción del código computacional del modelo de simulación basado en agentes del cuidado domiciliario. Con el fin de obtener los diagramas de clase se utilizó el complemento "ObjectAid UML Explorer" disponible para Eclipse. A continuación se presentan los diagramas de clase empleados en la construcción del modelo y se describen brevemente.

6.2.1. Clase AgenteGenerador

La clase AgenteGenerador implementa el comportamiento Generador el cual crea un archivo de texto plano mediante el método "Escribir" para cada periodo de la simulación con base en información histórica o aleatoria del sistema. El método Escribir llama los otros métodos de la clase que corresponden a los parámetros del sistema como se observa en la Figura 14. Una vez el archivo está creado envía un mensaje al Agente Coordinador en donde le indica que la información del periodo está disponible.

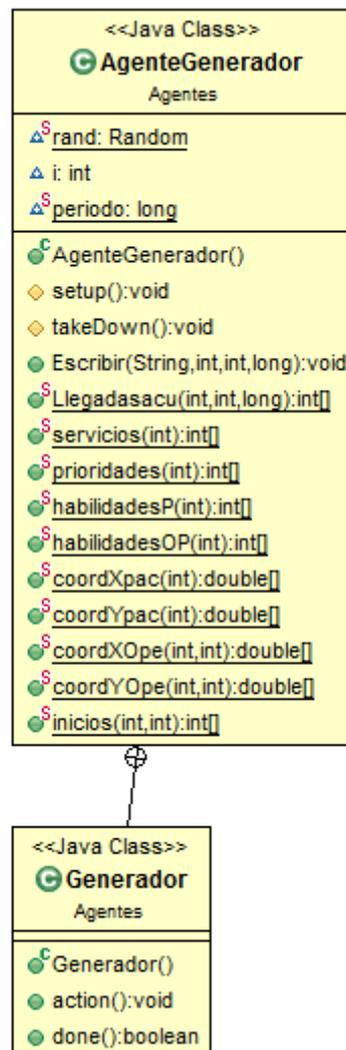


Figura 14 Diagrama de clase para la clase AgenteGenerador
Fuente: Autor

6.2.2. Clase AgenteCoordinador

La clase AgenteCoordinador ejecuta el comportamiento HacerRuteo la cual al el momento de recibir el mensaje del AgenteGenerador, se llama la clase LeerDatos la cual lee el archivo creado por el AgenteGenerador y almacena la información en las distintas variables del modelo. Una vez leída la información el AgenteCoordinador llama la clase Modelomatemat a la cual le envía la información del sistema para que en esa clase se resuelva el problema de ruteo para ese periodo, por último con la información del ruteo disponible el Agente Coordinador envía mensajes a los operadores de salud con las rutas y tiempos de llegada a los pacientes asignados a ellos. En la Figura 15 se presenta lo previamente descrito.

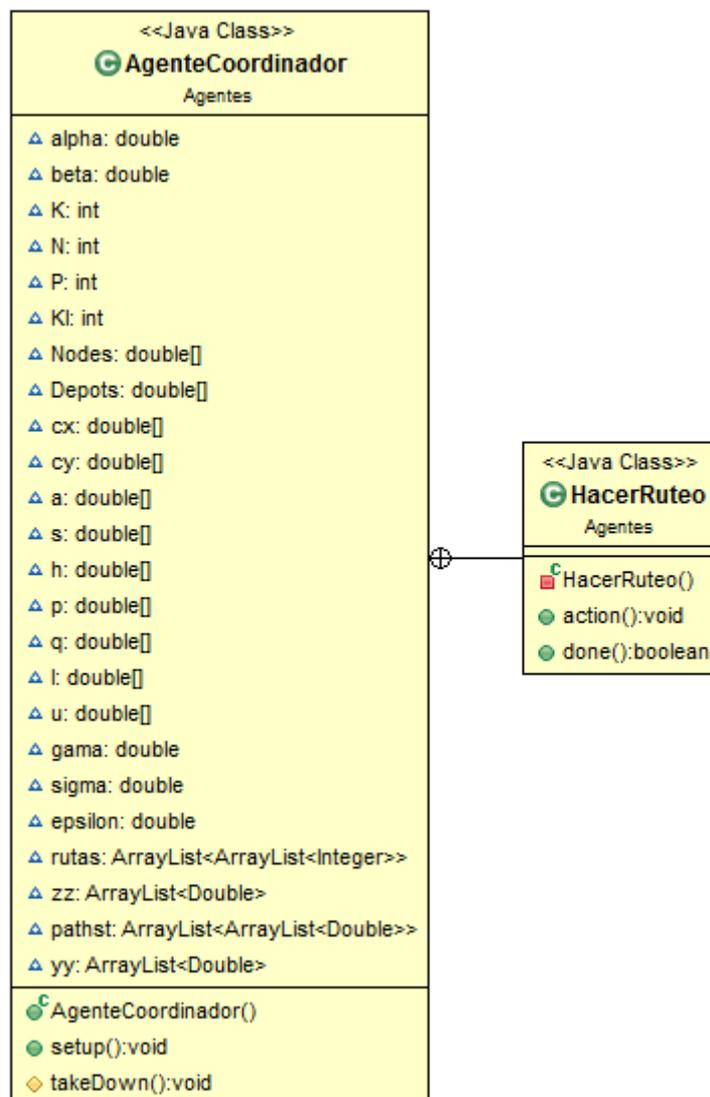


Figura 15 Diagrama de clase para la clase AgenteCoordinador
Fuente: Autor

6.2.3. Clase LeerDatos

La clase LeerDatos como se indicó anteriormente es utilizada por la clase AgenteCoordinador para acceder a la información del archivo de texto plano creado por el Agente Generador. Este archivo de texto cuenta con la información del periodo a realizar el ruteo por lo que se debe parametrizar la información con la cual se resolverá el problema. Para lograr lo anterior ésta clase almacena y devuelve la información del sistema en las variables que se encuentran en el diagrama de clase mostrado en la Figura 16. Es importante resaltar que en esta clase se utiliza la librería Java IO para acceder y leer archivos de texto plano, por otra parte esta clase no representa ningún agente en el sistema por lo que no tiene comportamientos.

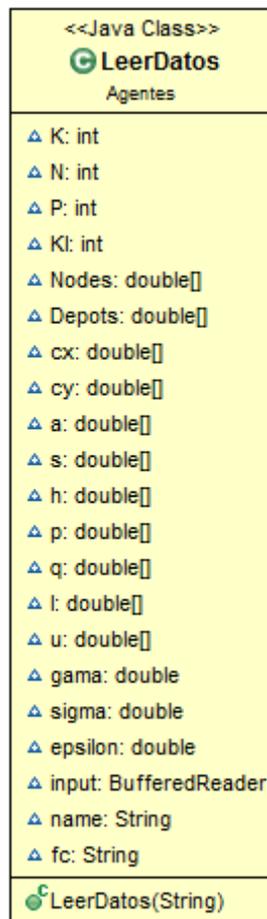


Figura 16 Diagrama de clase para la clase LeerDatos
Fuente: Autor

6.2.4. Clase Modelomatemat

La clase Modelomatemat como se indicó anteriormente es utilizada por el Agente Coordinador para realizar el ruteo de los operadores de salud. Es por ello que la clase como información de entrada recibe los parámetros del archivo de texto plano creado por el Agente Generador, resuelve el modelo matemático utilizando la librería de Xpress-MP y devuelve la solución óptima del ruteo en los arreglos correspondientes a las rutas de cada operador de salud, la promesa de servicio y el tiempo de llegada a cada paciente. En la Figura 17 se observa el diagrama de clases con las variables y objetos que se manejan.



Figura 17 Diagrama de clase para la clase Modelomatemat
Fuente: Autor

6.2.5. Clase AgenteOperador

La clase AgenteOperador representa a cada operador de salud que realizará las visitas domiciliarias. Los Agentes Operadores de salud reciben los esquemas de ruteo en un mensaje enviado por el Agente Coordinador con sus respectivas rutas y tiempos de llegada. La clase mediante el comportamiento AtenderPacientes imprime el listado de pacientes que visitará así como el tiempo en el que estará disponible de nuevo, además sobrescribe el archivo generado por el Agente Generador con el fin de notificar el tiempo en el cual volverán a estar disponible y su última ubicación después del ruteo. De esa manera se asegura que para el próximo esquema de ruteo del siguiente periodo su ubicación inicial sea la última ubicación del periodo anterior. En la Figura 18 se presenta el diagrama de clases para dos operadores de salud en donde se muestra la ejecución del comportamiento AtenderPacientes.

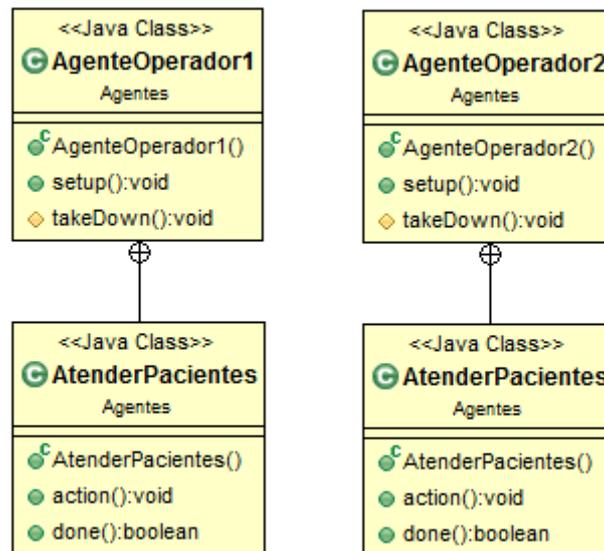


Figura 18 Diagrama de clase para la clase AgenteOperador
Fuente: Autor

Por último es importante resaltar que en el Anexo 3 se encuentra el código computacional de las clases de Java correspondientes al modelo de simulación basado en agentes.

6.3. SECUENCIA DE LA EJECUCIÓN DEL CÓDIGO COMPUTACIONAL

En este apartado de capítulo se profundizará en el funcionamiento del modelo de simulación basado en agentes mediante un paso a paso de todas las operaciones realizadas por el código computacional durante la simulación. Lo anterior es realizado con ayuda de capturas de pantalla de Eclipse así como de capturas de pantalla de los archivos de texto plano empleados en la ejecución.

1. Inicializar agentes

Jade mediante el método `getLocalName()` obtiene el nombre de los agentes que están activos e inicializados en el sistema que en este caso son el Agente Generador, el Agente Coordinador y los Agentes Operador de Salud. Por otra parte al principio de la simulación se imprime la información de la dirección del equipo en donde se está corriendo Jade lo cual es mostrado en la Figura 19.

```

INFORMACIÓN: MTP addresses:
http://JulianEspejoD:7778/acc
ago 06, 2016 10:18:04 PM jade.core.AgentContainerImpl joinPlatform
INFORMACIÓN: -----
Agent container Main-Container@192.168.0.4 is ready.
-----
Hola, soy el agenteAgenteOperador4
Hola, soy el agenteAgenteOperador2
Hola, soy el agenteAgenteOperador3
Hola, soy el agenteAgenteGenerador
Hola, soy el agenteAgenteCoordinador
Esperando a recibir aviso
Hola, soy el agenteAgenteOperador1
    
```

Figura 19 Inicialización de los Agentes
Fuente: Autor

2. Generar datos del periodo 1

El agente generador crea la información del sistema para el periodo 1 la cual es puesta en un archivo de texto llamado "Datos0.txt" como se observa en la Figura 20.

Nombre	Fecha de modifica...	Tipo
.settings	12/07/2016 22:45	Carpeta de arc...
bin	6/08/2016 22:17	Carpeta de arc...
src	12/07/2016 22:49	Carpeta de arc...
.classpath	16/07/2016 21:31	Archivo CLAS...
.project	3/08/2016 22:46	Archivo PROJ...
APDescription.txt	6/08/2016 22:18	Documento d...
Datos0.txt	6/08/2016 22:18	Documento d...
Datos1.txt	6/08/2016 22:18	Documento d...
Datos2.txt	6/08/2016 22:19	Documento d...
Datos3.txt	6/08/2016 22:19	Documento d...
Datos4.txt	6/08/2016 22:20	Documento d...
Datos5.txt	6/08/2016 22:20	Documento d...
Datos6.txt	4/08/2016 22:00	Documento d...

Datos									
K	4								
P	5								
gamma	16.071								
sigma	-37.929								
epsilon	24								
paciente									
1	!x	!y	!a	!s	!h	!p			
0	44.294	-46.364	0	0	0	0			
1	-57.552	-8.496	0	0	0	0			
2	-82.211	91.471	0	0	0	0			
3	-5.576	-43.926	0	0	0	0			
4	-17.351	45.226	5444	3623	3	3			
5	-42.655	-20.813	9834	3237	5	4			
6	24.924	-13.304	15168	3487	5	4			
7	46.292	58.712	20725	3680	2	1			
8	-2.793	17.784	25962	3525	3	1			
operadores									
K	q	l	u						
0	5	30000	800000						
1	5	30000	800000						
2	5	30000	800000						
3	5	30000	800000						

Figura 20 Datos para el primer periodo en la prueba del código
Fuente: Autor

3. Solución del problema de ruteo

Una vez generada la información el Agente Generador envía un mensaje tipo “ACL-INFORM” en donde le avisa al Agente Coordinador que la información para el periodo 1 está disponible, el Agente Coordinador lee la información en el archivo de texto “Datos0.txt” utilizando la clase *LeerDatos* y resuelve el problema de ruteo utilizando el modelo matemático del Capítulo 4 codificado en Java mediante la clase *Modelomatemat*. En la Figura 21 se observa el envío y recepción del mensaje del Agente Generador, además se muestra la manera en que se ejecuta la librería de optimización de Xpress en Java con el fin de resolver el problema.

```
Enviando información del sistema de cuidado a la salud domiciliaria para el periodo 0-30000
Datos generados, Leyendo información
Generando ruteo de la iteración: 1
Read Data Time 0.0 seg

Start of Routing Model

Reading Problem Routing
Problem Statistics
    646 (    0 spare) rows
    441 (    0 spare) structural columns
    2369 (    0 spare) non-zero elements
Global Statistics
    384 entities          0 sets          0 set members
Minimizing MILP Routing
Original problem has:
    646 rows          441 cols          2369 elements          384 globals
Presolved problem has:
    193 rows          149 cols          828 elements          124 globals
LP relaxation tightened
Will try to keep branch and bound tree memory usage below 4.1Gb
Starting concurrent solve with dual

Concurrent solve, 0s
Dual
  objective  dual inf
  dual crash
D 8511.4467  .0000000
P 8685.4446  .0000000
----- optimal -----
Optimal solution found
Dual solved problem
  65 simplex iterations in 0s
```

Figura 21 Solución del problema de ruteo en Java
Fuente: Autor

4. Distribuir del esquema de ruteo a los agentes operador de salud

Una vez obtenido el esquema de ruteo óptimo para el periodo 1 se envía a cada operador de salud por separado, para ello el Agente Coordinador mediante la utilización de arreglos dinámicos separa la información para cada agente y luego la envía a través de un mensaje tipo “ACL INFORM”. La información enviada en el mensaje corresponde a la lista de pacientes a los cuales el operador de salud debe visitar y el tiempo de llegada a los hogares de los pacientes.

5. Realizar las visitas domiciliarias y calcular el tiempo de disponibilidad del operador de salud

Los Agentes Operador de Salud reciben el mensaje por parte del Agente Coordinador con el esquema de ruteo y los tiempos de llegada y realizan las visitas domiciliarias. Posterior a eso imprimen el listado de pacientes que visitaron así como el tiempo en el que estarán disponibles de nuevo el cual equivale al tiempo de llegada al último paciente de la ruta recibida más el tiempo de servicio de ése paciente. En la Figura 22 se observa lo anteriormente descrito.

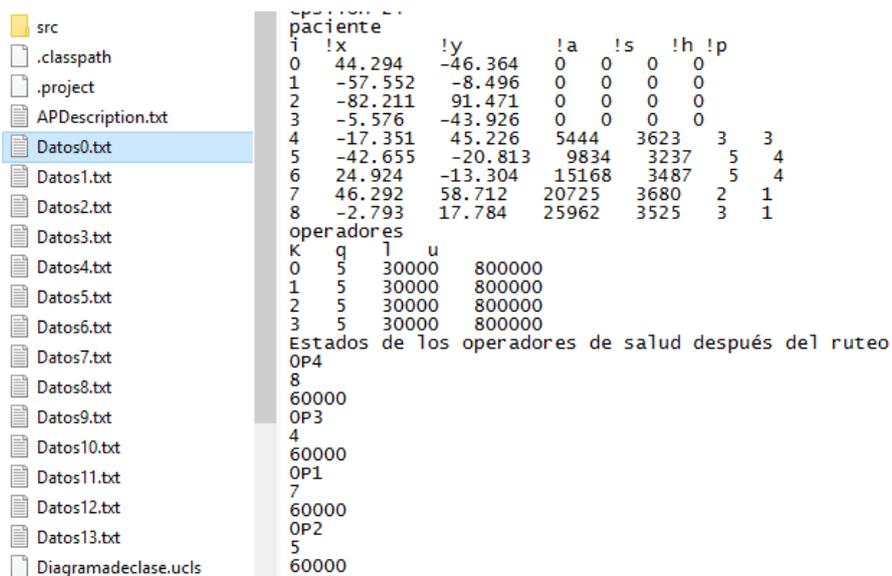
```

Number of integer feasible solutions found is 11
Best integer solution found is 15718.06160
Best bound is 15717.66068
Uncrunching matrix
Objective: 15718.0616
Computational Time 2.024 seg
Esperando a recibir aviso
Operador 2 va a los nodos [5.0] llega en el tiempo [30027.214000000007] respectivamente y estará disponible en: 33264
Operador 3 va a los nodos [4.0] llega en el tiempo [30111.1049999999996] respectivamente y estará disponible en: 33734
Operador 1 va a los nodos [6.0, 7.0] llega en el tiempo [30052.4299999999993, 33632.8140000000006] respectivamente y estará disponible en: 37312
Operador 4 va a los nodos [8.0] llega en el tiempo [30064.4930000000002] respectivamente y estará disponible en: 33589
    
```

*Figura 22 Pacientes a visitar, tiempos de llegada y disponibilidad de cada operador
Fuente: Autor*

6. Notificar la ubicación y el tiempo en el cual vuelven a estar disponibles los operadores de salud

Los operadores de salud deben notificar al Agente Generador su ubicación y el tiempo en el cual vuelven a estar disponibles, para ello se sobrescribe el archivo de texto plano "Datos0.txt" en donde en las últimas líneas colocan el número del nodo en el cual se encuentran además de colocar el tiempo en el cual vuelven a estar disponibles. Es importante resaltar que si los operadores de salud terminan antes del comienzo del próximo periodo, colocan el tiempo de comienzo del próximo periodo, pues es el momento en el cual conocen la información del siguiente ruteo y es el tiempo con el cual resuelven el modelo matemático.



*Figura 23 Notificación del tiempo y momento de disponibilidad
Fuente: Autor*

En la Figura 23 se muestra que se sobrescribió el archivo “Datos0.txt” agregando en las últimas líneas para cada operador el nodo en el cual quedaron además del tiempo desde donde empiezan el próximo esquema de ruteo. Por ejemplo el operador de salud 1 va a los pacientes de los nodos 6 y 7 y termina el ruteo y el servicio en el tiempo 37312, por lo que en las últimas líneas está el identificador del operador de salud 1 (OP1), la siguiente línea está el nodo 7 que es donde terminó el ruteo y en la siguiente línea el número 60000 que es cuando comienza el próximo ruteo.

7. Generar los datos del periodo 2 en adelante

Para los siguientes periodos el Agente Coordinador debe utilizar la información de la ubicación y el tiempo de disponibilidad del periodo anterior. Por ejemplo para el periodo 2 debe utilizar los datos del periodo 1 que están en el archivo “Datos0.txt”, para los datos del periodo 2 debe utilizar la información del archivo “Datos1.txt” y así para los demás periodos. La información que toma de éstos archivos planos de texto es la escrita por los operadores de salud en el periodo anterior con el número del nodo en el cual quedó el operador de salud el agente coordinador busca las coordenadas de ése nodo y las escribe en el nodo de donde empieza el operador de salud en el siguiente periodo.

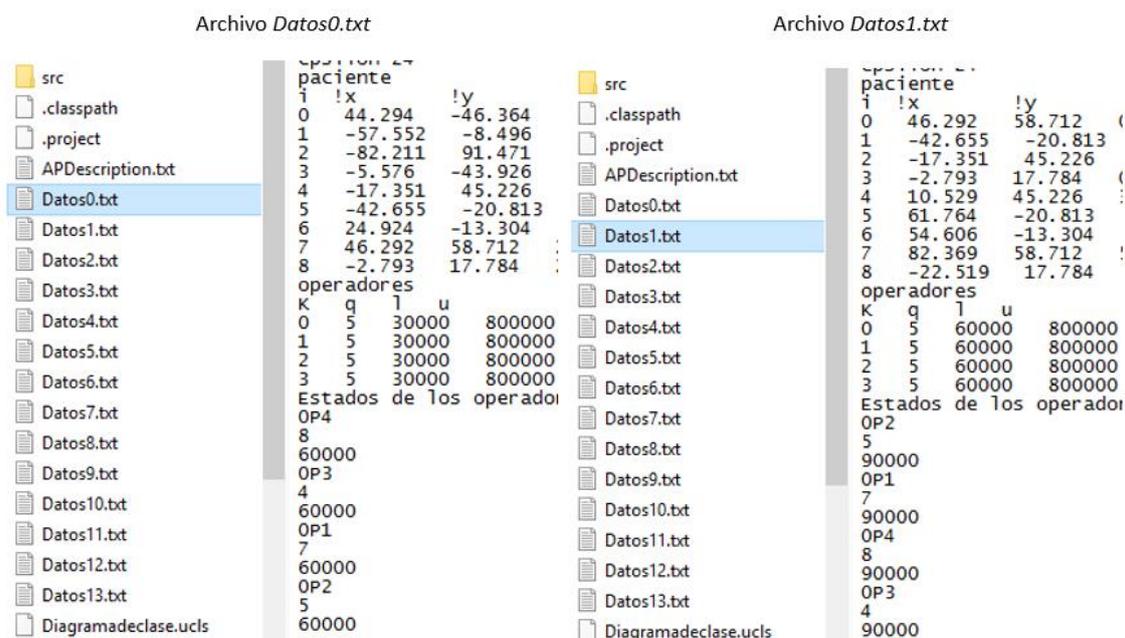


Figura 24 Generación de datos del periodo 2 en adelante

Fuente: Autor

En la Figura 24 se observa lo descrito anteriormente, pues en el archivo “Datos0.txt” se muestra que el operador de salud 1 terminó en el nodo 7 que está en las coordenadas X=46.292, Y=58.712 las cuales las escribe en el archivo “Datos1.txt” en el nodo 0 que es la ubicación de inicio del operador de salud 1 para el ruteo del periodo 2. De la misma forma es realizado para los otros operadores de salud, por ejemplo el operador 2 terminó en las coordenadas del nodo 5 las cuales van a servir de iniciación del operador de salud dos que inicia en el nodo 1 para el periodo 2. Por otra parte el momento de inicio de disponibilidad se actualiza en “Datos1.txt” a 60000 como está escrito en el archivo “Datos0.txt”.

Después de generados los datos para el periodo dos se procede a realizar el ruteo ahora con los pacientes que ingresaron en éste periodo. Para ello se vuelve al paso 3 y se repite hasta completar los periodos de simulación del horizonte de planeación.

6.4. PRUEBAS DEL CÓDIGO COMPUTACIONAL

Con el fin de establecer el alcance de la implementación del modelo de simulación basado en agentes se realizan distintas pruebas variando la configuración para cada prueba. El sistema se configura con parámetros aleatorios los cuales buscan que el código sea probado en diferentes circunstancias.

Para realizar las pruebas de lo construido en Java se plantea atender 40 pacientes durante un horizonte de planeación de 600000 unidades de tiempo que corresponden a 600 minutos con pacientes que ingresan al sistema cada 15 minutos. En las diferentes pruebas de ejecución se varía la duración de cada periodo de simulación, el número de pacientes a ingresar por periodo, el número de operadores de salud en el sistema, y finalmente se varían los parámetros α y β del modelo matemático utilizado para realizar el ruteo. Todo lo anterior se varía siempre buscando atender 40 pacientes en 600 minutos de simulación en los cuales ingresan pacientes.

Es importante aclarar que para todas las pruebas y ejecuciones del modelo se utilizó la misma semilla a la hora de generar números aleatorios. Lo anterior es realizado con el fin de que se generen los mismos valores para las coordenadas de los nodos, las prioridades, las habilidades requeridas y los tiempos de servicio para cada paciente. Un ejemplo de lo anterior es que para todas las pruebas el paciente 1 estará en la misma ubicación, requerirá de la misma habilidad del operador de salud, el tiempo que requiere para su atención es el mismo y su prioridad es igual para todas las pruebas. A continuación se presentan las pruebas con sus respectivas experimentaciones y el análisis de los resultados para cada prueba.

6.4.1. Prueba 1 – Experimentación y resultados

Para la primera prueba el sistema se configura utilizando periodos de simulación de 120 minutos en los cuales entran 8 solicitudes por periodo como se muestra en la Figura 25 para un total de 40 pacientes en 600 minutos de simulación. Los demás parámetros del modelo de simulación se encuentran en la Tabla 21.

IMPLEMENTACIÓN DEL SISTEMA MULTI-AGENTE DE CUIDADO DOMICILIARIO

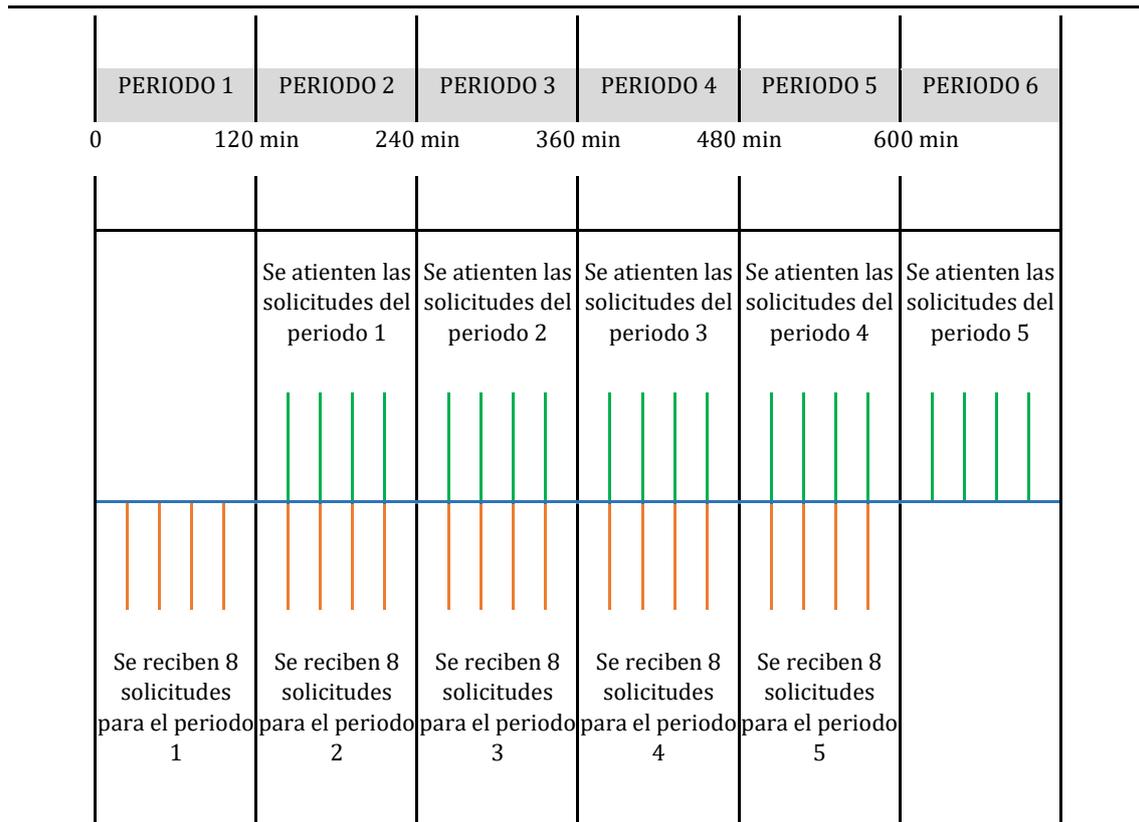


Figura 25 Configuración de la prueba de ejecución 1
Fuente: Autor

Tabla 21 Parámetros de la prueba de ejecución 1
Fuente: Autor

Parámetro	Símbolo	Configuración
Horizonte de planeación	HP	600 minutos
Duración de cada periodo	i	120 minutos
Número de periodos	ni	5
Número de pacientes por periodo	P	8
Total de pacientes a atender	TP	40
Número de operadores de salud	K	4
Tiempo entre llegada de pacientes	a_i	15 minutos
Tiempo de servicio al paciente	s_i	Uniforme (8,10) minutos
Prioridades en la atención a los pacientes	p_i	Uniforme Discreta (2,4)
Habilidad requerida por los pacientes	h_i	Uniforme Discreta (1,3)
Habilidades de los operadores de salud*	q_k	5 en todos los casos
Coordenada en X de los pacientes	xx_i	Uniforme (-100,100)
Coordenada en Y de los pacientes	yy_i	Uniforme (-100,100)
Coordenada en X del hogar del operador de salud	xx_k	Uniforme (-100,100)
Coordenada en Y del hogar del operador de salud	yy_k	Uniforme (-100,100)
Comienzo de la jornada laboral	l_k	Minuto 120
Finalización de la jornada laboral	u_k	Minuto 720

MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA

Es importante resaltar que para cada prueba se realizan diferentes corridas de la simulación variando los parámetros α y β del modelo matemático. En la Tabla 22 se encuentran los resultados de las experimentaciones de la prueba 1 en donde FO hace referencia al valor total de la función objetivo, FO1 a la función de ruteo y FO2 a la función de promesa de servicio. Por otra parte se utilizan los valores acumulados de las funciones objetivo y por otra parte los tiempos promedio de espera debido a que en una misma experimentación se resuelven tantos modelos de optimización como periodos tiene la simulación (5 periodos en este caso). En otras palabras, en las casillas correspondientes a las funciones objetivo se encuentra la sumatoria de los resultados de los 5 modelos de optimización resueltos en la simulación.

Tabla 22 Resultados de la prueba de ejecución 1
Fuente: Autor

α	β	FO Acumulada	FO₁ Acumulada	FO₂ Acumulada	Tiempo de espera promedio (min)
1	0	3115.024	3115.024	0	89.270
0.8	0.2	3264.473	3113.0488	3870.17	79.417
0.6	0.4	3395.076	3103.4933	3832.45	78.24
0.4	0.6	3471.326	3008.9525	3779.575	70.005
0.2	0.8	3552.874	3185.96	3644.603	69.555

En ésta prueba se observa que el tiempo promedio de espera disminuye con un $\alpha=0.2$ y $\beta=0.8$ debido a que la función objetivo de promesa de servicio tiene más importancia y por lo tanto se utilizan más operadores de salud para atender a los pacientes disminuyendo el tiempo de espera promedio. Por otra parte se muestra que al disminuir el tiempo promedio de espera la FO Acumulada aumenta. La experimentación con $\alpha=0$ y $\beta=1$ no fue posible realizarla debido a que el tiempo computacional requerido para realizar el ruteo con éstos parámetros, debido a que se excede el tiempo del periodo de simulación generando errores en la ejecución.

6.4.2. Prueba 2 – Experimentación y resultados

Para la prueba de ejecución número 2 se utilizan los mismos parámetros que la prueba número 1 disminuyendo el número de operadores de salud de 4 a 2.

Tabla 23 Resultados de la prueba de ejecución 2
Fuente: Autor

α	β	FO Acumulada	FO₁ Acumulada	FO₂ Acumulada	Tiempo de espera promedio (min)
1	0	3179.556	3179.556	0	82.227
0.8	0.2	3344.295	3209.6913	3882.71	80.771
0.6	0.4	3478.901	3209.6917	3882.715	77.954
0.4	0.6	3528.116	3099.23	3814.04	73.737
0.2	0.8	3654.445	3179.705	3773.13	69.812

En la Tabla 23 se muestra que al disminuir a la mitad los operadores de salud se aumenta el tiempo promedio de espera en el sistema en un 2.3% lo que indica que con dos operadores de salud se puede realizar la atención de los pacientes obteniendo similares tiempos de espera e incurriendo en menores costos de contratación y de salarios de los operadores de salud en el sistema.

6.4.3. Prueba 3 – Experimentación y resultados

Como se indicó anteriormente las experimentaciones del modelo de simulación buscan atender a 40 pacientes en un periodo de simulación de 600 minutos. En esta prueba se manejan 8 periodos de 75 minutos en los cuales ingresan 5 pacientes por periodo para un total de 40 pacientes en un tiempo total de 600 minutos. Lo anterior se muestra en la Figura 26 y es complementado con los valores de la Tabla 24 que corresponden a los demás parámetros del modelo los cuales son iguales a los de la prueba 1. Igualmente se realizan experimentaciones del modelo variando los parámetros α y β , por otra parte en ésta oportunidad el valor de las funciones objetivo para cada α y β se calcula obteniendo valor acumulado de las funciones objetivo de los ruteos realizados, es decir, como se manejan 8 periodos de 75 minutos para cada α y β los valores de las casillas de función objetivo se calculan acumulando los resultados de los 8 ruteos resueltos. Lo anterior es presentado en la Tabla 25.

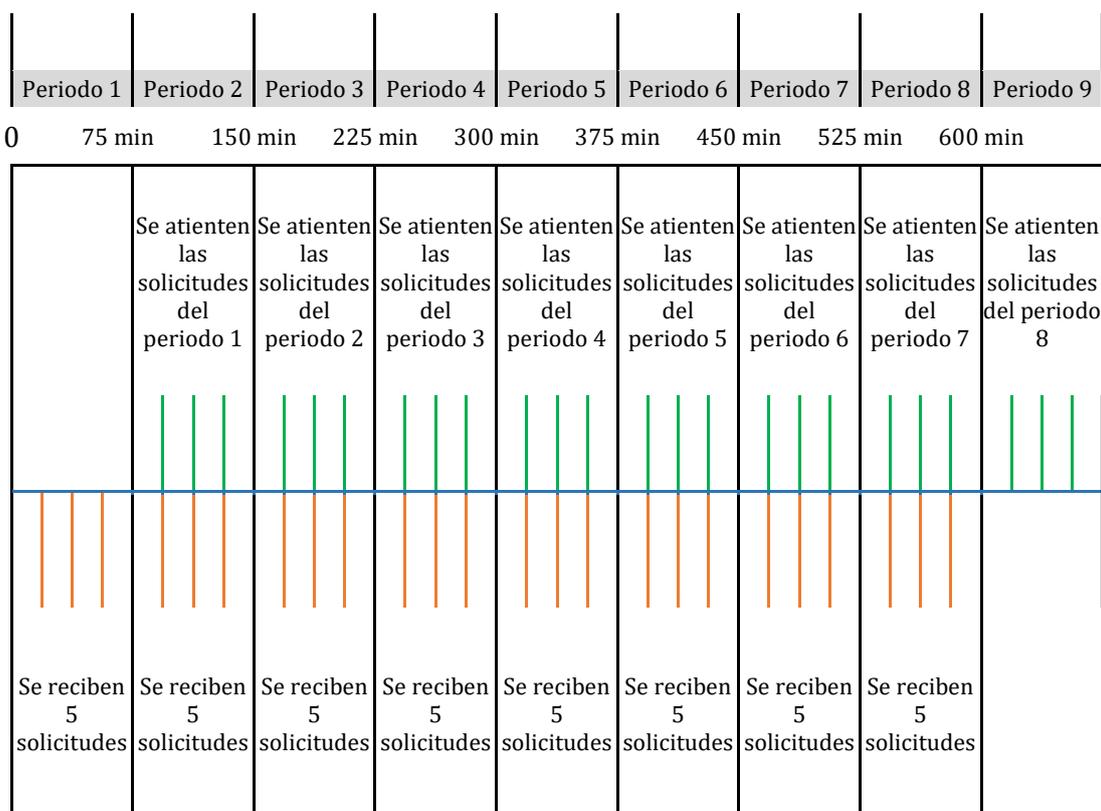


Figura 26 Configuración de la prueba de ejecución 3
Fuente: Autor

MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA

Tabla 24 Parámetros de la prueba de ejecución 2

Fuente: Autor

Parámetro	Símbolo	Configuración
Horizonte de planeación	HP	600 minutos
Duración de cada periodo	i	75 minutos
Número de periodos	ni	8
Número de pacientes por periodo	P	5
Número de operadores de salud	K	4
Tiempo entre llegada de pacientes	a_i	15 minutos
Tiempo de servicio al paciente	s_i	Uniforme (8,10) minutos
Prioridades en la atención a los pacientes	p_i	Uniforme Discreta (1,5)
Habilidad requerida por los pacientes	h_i	Uniforme Discreta (1,5)
Habilidades de los operadores de salud*	q_k	5 en todos los casos
Coordenada en X de los pacientes	xx_i	Uniforme (-100,100)
Coordenada en Y de los pacientes	yy_i	Uniforme (-100,100)
Coordenada en X del hogar del operador de salud	xx_k	Uniforme (-100,100)
Coordenada en Y del hogar del operador de salud	yy_k	Uniforme (-100,100)
Comienzo de la jornada laboral	l_k	Minuto 75
Finalización de la jornada laboral	u_k	Minuto 675

Tabla 25 Resultados de la prueba de ejecución 3

Fuente: Autor

α	β	FO Acumulada	FO_1 Acumulada	FO_2 Acumulada	Tiempo de espera promedio (min)
1	0	3527.26	3527.26	0	39.410
0.8	0.2	3863.09	3449.98	5515.5	37.519
0.6	0.4	4276.19	3449.98	5515.5	37.343
0.4	0.6	4688.7	3470.7	5500.7	36.293
0.2	0.8	5110.72	3520.13	5508.37	36.770
0	1	5515.65	0	5515.65	31.773

En los resultados de ésta prueba se observa que el tiempo promedio de espera de los pacientes para recibir el servicio disminuye conforme se aumenta la importancia de la promesa de servicio en el sistema, es decir, aumentando β . En contraste si α tiene mayor importancia, es decir, el tiempo de ruteo tiene mayor importancia el tiempo de espera promedio aumentará. Un balance entre la función objetivo de ruteo y de promesa de servicio se encuentra con $\alpha=0.6$ y $\beta=0.4$ sin sacrificar en mayor medida el tiempo de espera promedio. Es criterio del proveedor del servicio establecer cual función objetivo tiene mayor importancia para el sistema de cuidado a la salud domiciliaria.

6.4.4. Prueba 4– Experimentación y resultados

Para la prueba de ejecución número 4 se utilizan los mismos parámetros que la prueba número 3 disminuyendo el número de operadores de salud de 4 a 2.

Tabla 26 Resultados de la prueba de ejecución 4
Fuente: Autor

α	β	<i>FO</i> Acumulada	<i>FO</i> ₁ Acumulada	<i>FO</i> ₂ Acumulada	Tiempo de espera promedio (min)
1	0	4754.88	4754.88	0	59.775
0.8	0.2	4931.1	4529.17	6538.8	44.304
0.6	0.4	4969.09	4529.17	5628.97	44.311
0.4	0.6	5183.07	4559.47	5598.8	49.924
0.2	0.8	5384.99	4681.38	5560.9	40.276
0	1	5515.65	0	5515.65	37.465

En los resultados de ésta prueba presentados en la Tabla 26 se resalta que el tiempo promedio de espera del sistema aumenta un 26% utilizando 2 operadores de salud en vez de 4 operadores de salud como se realizó en la prueba 3. Lo anterior evidencia que si se requiere disminuir el tiempo de espera promedio en el sistema se deben emplear más operadores de salud.

6.5. CONCLUSIONES DEL CAPÍTULO

En éste capítulo se presentó la implementación del sistema multi-agente en Java utilizando la librería de agentes JADE. Para lograr lo anterior se definió el software a utilizar especificando las librerías empleadas. Posteriormente se presentaron los diagramas de clases de la implementación en Java. Por otra parte se realizaron pruebas de ejecución del código con parámetros aleatorios con el fin de evaluar el desempeño de lo construido en Java para finalmente mostrar y analizar los resultados a los que se llegaron en la prueba de ejecución.

Se concluye que la implementación del sistema multi-agente está de acuerdo con lo planteado en el capítulo 5. Lo anterior es a causa de que en la implementación computacional se cumple con los protocolos de comunicación, con las actividades de desarrolladas por cada agente, con las interacciones entre los agentes y con las reglas generales del sistema de cuidado domiciliario. Adicional a lo anterior la implementación es capaz de generar la información de entrada y llegar a resultados coherentes haciendo uso de la configuración del sistema así como del modelo matemático para realizar el ruteo.

Se evidencia que los pacientes son visitados una sola vez y el tiempo de espera disminuye si la solicitud se realiza cerca al momento en que se termina el periodo de recepción de solicitudes lo cual refleja lo planteado en la aproximación multi-agente. Por otra parte se observa que los operadores de salud no se regresan a sus hogares sino que permanecen en las ubicaciones de los pacientes hasta recibir un nuevo esquema de ruteo, en caso de que no reciban información y se termine el horizonte de planeación retornan a sus hogares según lo establecido en las reglas del sistema.

Adicional a lo anterior se concluye que haciendo más corto el periodo de simulación en las ejecuciones se disminuye el tiempo de espera de los pacientes para recibir el servicio. Lo anterior se muestra en el tiempo promedio de espera de las pruebas 1 y 3, debido a que para la prueba 1 fue de 64.414 minutos con un periodo de simulación de 120 minutos, mientras que en la prueba 2 con un periodo de simulación de 75 minutos el tiempo promedio de espera en el sistema es de 36.518 minutos. Sin embargo la realización de periodos de simulación más cortos implica mayores costos para el sistema lo que se muestra en los valores de las funciones objetivo para las pruebas 1 y 3.

El número de operadores de salud también influye en el desempeño del modelo de simulación basado en agentes. En la prueba 4 se utilizó la misma configuración y los mismos parámetros que en la prueba 3 disminuyendo el número de operadores de salud de 4 a 2, dando como resultado un aumento del 25.99% en el tiempo promedio de espera de los pacientes para recibir el servicio.

Finalmente se concluye que los periodos de simulación deben ser lo suficientemente amplios para permitir al optimizador llegar a la solución del problema. En caso de que el optimizador no alcance a arrojar las rutas y los tiempos de llegada de los operadores de salud se generan arreglos vacíos los cuales generan errores en la ejecución.

6.6. RESULTADOS DE INVESTIGACIÓN

El contenido de éste capítulo junto con el capítulo 5 fue presentado y aceptado para presentación en el evento “WORKSHOP ON ENGINEERING APLICATIONS 2016” a través del artículo: E. López-Santana, J. Espejo-Díaz, and G. Méndez-Giraldo, “Multi-agent approach for solving the dynamic home health care routing problem”.

Capítulo 7

ESTUDIO DE CASO

En éste capítulo se realiza el estudio de caso con información real de un proveedor del servicio de cuidado domiciliario mediante una simulación basada en agentes. En el estudio de caso se presenta la configuración y los parámetros de la ejecución del modelo basado en agentes, se realizan experimentaciones con un número determinado de operadores de salud disponibles en el horizonte de planeación para finalmente analizar los resultados del caso de estudio. Es importante resaltar que el estudio de caso se realiza con el objetivo de mostrar la configuración y el ingreso de parámetros reales al modelo implementado en Java realizando una breve experimentación del sistema.

7.1. DESCRIPCIÓN DE LA SITUACIÓN

Un proveedor del servicio de cuidado a la salud domiciliaria cuenta con un total de 21319 usuarios afiliados al servicio los cuales se distribuyen por localidades como se muestra en la Tabla 27. Los usuarios al momento de notificar su necesidad del servicio se convierten en pacientes a los cuales se les asigna y envía un operador de salud calificado para que realice la atención correspondiente. El proveedor del servicio debe determinar el número de operadores de salud a tener en el sistema así como las rutas que éstos deben seguir con el fin de atender a los pacientes teniendo en cuenta la prioridad en la atención y el tiempo de transporte de los operadores de salud.

Tabla 27 Usuarios del servicio de cuidado domiciliario por localidad
Fuente: Autor

Número de localidad	Localidad	Número de usuarios
1	Usaquén	2917
2	Chapinero	886
3	Santafé	402
4	San Cristóbal	220
5	Usme	73
6	Tunjuelito	174
7	Bosa	232
8	Kennedy	2195
9	Fontibón	1950
10	Engativá	3008
11	Suba	4881
12	Barrios unidos	1013

13	Teusaquillo	1355
14	Los mártires	283
15	Antonio Nariño	263
16	Puente Aranda	748
17	La Candelaria	200
18	Rafael Uribe	228
19	Cuidad Bolívar	156
20	Comuna 1	34
21	Comuna 2	32
22	Comuna 5	51
23	Comuna 6	18

Por otra parte el proveedor del servicio de cuidado domiciliario cuenta con información histórica del sistema la cual recolectó durante 92 días consecutivos. A continuación se lista la información recolectada del sistema de cuidado a la salud domiciliaria, además de información con que cuenta el proveedor.

- Información histórica de las solicitudes por hora presentadas en 92 días.
- Georreferenciación o ubicación espacial de los hogares de los usuarios mediante las coordenadas de latitud y longitud.
- Clasificación de los usuarios del sistema según la localidad en donde viven.
- Frecuencia de solicitudes por localidad.

Finalmente y considerando que no se cuenta con la totalidad de la información necesaria para configurar el sistema, se realizan las siguientes suposiciones para configurar el sistema.

- Georreferenciación de los hogares de los operadores de salud.
- Tiempo de servicio de los pacientes.
- Nivel de habilidad de los operadores de salud.
- Nivel de habilidad requerido por los pacientes para que se les realice una visita domiciliaria.
- Nivel de prioridad en la atención de los pacientes.

A continuación se presenta el análisis de entrada de la información con que se dispone del sistema, para posteriormente establecer la configuración de la simulación basada en agentes con base en la información de entrada y los supuestos realizados.

7.2. INFORMACIÓN Y ANÁLISIS DE ENTRADA DEL SISTEMA

En este apartado de capítulo se muestra el resultado del análisis de entrada de la información disponible sobre el sistema de cuidado domiciliario. Por otra parte se presenta información relevante con que cuenta el proveedor del servicio que es necesaria para la configuración de la ejecución.

7.2.1. Análisis de entrada de las solicitudes por hora del sistema

Como se indicó anteriormente se cuenta con información histórica de las solicitudes del servicio por horas (ver anexo 4). La anterior información se analizó con el software Stat::Fit de Promodel obteniendo la función de distribución de llegada de pacientes al sistema por hora que se encuentra en la Tabla 28. Las pruebas del análisis de entrada se encuentran en el Anexo 5 del presente documento.

*Tabla 28 Distribuciones de solicitudes por horas
Fuente: Autor*

(0-1)	Binomial (2,0.103261)
(1-2)	Geométrica (0.8288)
(2-3)	Geométrica (0.8363)
(3-4)	Geométrica (0.8598)
(4-5)	Poisson(0.4130)
(5-6)	Poisson(0.8804)
(6-7)	Geométrica (0.3216)
(7-8)	Poisson(2.4347)
(8-9)	Poisson(2.6521)
(9-10)	Poisson(2.1630)
(10-11)	Poisson(1.8695)
(11-12)	Poisson(1.4239)
(12-13)	Geométrica (0.3622)
(13-14)	Poisson(1.3913)
(14-15)	Poisson(1.3152)
(15-16)	Poisson(1.6413)
(16-17)	Poisson(2.0326)
(17-18)	Poisson(2.0869)
(18-19)	Poisson(2.3043)
(19-20)	Poisson(2.1087)
(20-21)	Poisson(1.5217)
(21-22)	Poisson(0.9565)
(22-23)	Poisson(0.5543)
(23-24)	Poisson(0.2391)

7.2.2. Frecuencia de solicitudes por localidad

Por otra parte el proveedor del servicio ha registrado la frecuencia de las solicitudes que ingresan al sistema por localidad. Lo anterior es de utilidad para el modelo debido a que utilizando un número aleatorio el sistema toma en cuenta las frecuencias por localidad presentadas en la Tabla 29.

Tabla 29 Frecuencia de solicitudes por localidad

Número de Localidad	Localidad	Frecuencia de Solicitudes
1	Usaquén	0.112200594
2	Chapinero	0.036483486
3	Santafé	0.012795203
4	San Cristóbal	0.014052394
5	Usme	0.002085338
6	Tunjuelito	0.008087521
7	Bosa	0.016769146
8	Kennedy	0.140612362
9	Fontibón	0.10241727
10	Engativá	0.178701423
11	Suba	0.193143053
12	Barrios unidos	0.046538566
13	Teusaquillo	0.051581554
14	Los mártires	0.013739318
15	Antonio Nariño	0.010807053
16	Puente Aranda	0.025062617
17	La Candelaria	0.005531787
18	Rafael Uribe	0.013998612
19	Ciudad Bolívar	0.010285773
20	Comuna 1	0.001136874
21	Comuna 2	0.001118869
22	Comuna 5	0.001714313
23	Comuna 6	0.001136874
	TOTAL	1

7.2.3. Georreferenciación de los pacientes.

Por último el proveedor del servicio cuenta con la ubicación geográfica de todos los usuarios en la ciudad de Bogotá a través de sus coordenadas geográficas (latitud y longitud). Con el fin de manejar la información esta se filtró y se almacenó por localidad en archivos de texto plano en los cuales el nombre corresponde al número de la localidad, por ejemplo, en el archivo de texto "1.txt" se encuentra la georreferenciación de los usuarios del sistema que viven en la localidad 1 que corresponde a Usaquén. En la Figura 27 se muestra el archivo de texto "1.txt" en donde se encuentra el identificador de la Localidad, el identificador del paciente y su respectiva latitud y longitud.

	Id Localidad	Id Paciente	Longitud	Latitud
JULIANE (F:) > DATOS PLANOS COORDENADAS				
Nombre				
1.txt	1-1	-74.0354922	-4.73507047	
2.txt	1-2	-74.0354922	-4.73507047	
3.txt	1-3	-74.0354922	-4.73507047	
4.txt	1-4	-74.03270686	-4.73323183	
5.txt	1-5	-74.0335413	-4.72898795	
6.txt	1-6	-74.03993038	-4.72985841	
7.txt	1-7	-74.03015248	-4.72186303	
8.txt	1-8	-74.03535756	-4.73005191	
9.txt	1-9	-74.03622112	-4.69394497	
10.txt	1-10	-74.04374177	-4.76451886	
	1-11	-74.0440177	-4.74511244	
	1-12	-74.02659654	-4.76237971	
	1-13	-74.03535756	-4.73005191	
	1-14	-74.03535756	-4.73005191	
	1-15	-74.04768501	-4.72455886	
	1-16	-74.03305146	-4.76995556	
	1-17	-74.0323942	-4.74458917	
	1-18	-74.03827361	-4.70067648	
	1-19	-74.03827361	-4.70067648	

Figura 27 Coordenadas de los pacientes por localidades

7.3. CONFIGURACIÓN DE LA SIMULACIÓN BASADA EN AGENTES

7.3.1. Calculo de las distancias y tiempos de desplazamiento entre nodos

Partiendo del hecho de que la información de entrada de las ubicaciones de los pacientes está en forma de coordenadas geográficas (latitud y longitud), el cálculo de las distancias y el posterior cálculo del tiempo de traslado debe ser realizado de una forma diferente.

Por una parte para las distancias entre los nodos se utilizó la fórmula de aproximación encontrada en [105] la cual corresponde a las ecuaciones 26 a 28.

$$d = \sqrt{x^2 + y^2} \quad (26)$$

En donde:

$$x = 110.56 (Latitud_j - Latitud_i) \quad (27)$$

$$y = 84.8(Longitud_j - Longitud_i) \quad (28)$$

Con la distancia entre los vértices se procede a calcular el tiempo de desplazamiento utilizando el factor de velocidad promedio de la ciudad de Bogotá que corresponde a 29.09 k/h según la publicación “MOVILIDAD EN CIFRAS 2014” de la secretaría de movilidad Distrital disponible en [106]. Es importante resaltar que con las anteriores ecuaciones se calcula la distancia esférica entre dos coordenadas sin tener en cuenta la distancia de las calles que es la que un vehículo debe recorrer por lo tanto es la real. En la Figura 28 tomada de [107], se observa la diferencia entre la distancia esférica que no toma en cuenta la malla vial y la distancia real que si la toma en cuenta y por lo tanto la distancia real es mayor a la distancia esférica.



Figura 28 Distancia esférica y distancia real
Fuente: Tomado de [107]

7.3.2. Parámetros del modelo de simulación

Como se observó anteriormente, se cuenta con información de entrada de pacientes cada hora, por lo que los periodos de simulación corresponderán a 60 minutos para un total de 24 periodos que corresponden a un día de ejecución. Los demás parámetros del modelo son presentados en la Tabla 30.

Tabla 30 Parámetros del estudio de caso
Fuente: Autor

Parámetro	Símbolo	Configuración
Horizonte de planeación	HP	1440 minutos
Duración de cada periodo	i	60 minutos
Número de periodos	ni	24
Número de pacientes por periodo	P	Según distribución Tabla 27
Total de pacientes a atender	TP	Según distribución Tabla 27
Número de operadores de salud	K	4
Tiempo entre llegada de pacientes	a_i	Uniforme (10-20)
Tiempo de servicio al paciente	s_i	Uniforme (15,20) minutos
Prioridades en la atención a los pacientes	p_i	Uniforme Discreta (2,4)
Habilidad requerida por los pacientes	h_i	Uniforme Discreta (2,5)

MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA

Habilidades de los operadores de salud*	q_k	Según Tabla 28
Coordenada en X de los pacientes	xx_i	Según Tabla 32
Coordenada en Y de los pacientes	yy_i	Según Tabla 32
Coordenada en X del hogar del operador de salud	xx_k	Según Tabla 31
Coordenada en Y del hogar del operador de salud	yy_k	Según Tabla 31
Comienzo de la jornada laboral	l_k	Minuto 60
Finalización de la jornada laboral	u_k	Minuto 1500

Es importante resaltar que no se dispone de la información de la ubicación de los hogares de los operadores de salud por lo que en la se supone ésta información ubicando a los operadores de salud aleatoriamente repartidos en la ciudad de Bogotá. En la Tabla 31 se encuentran las coordenadas geográficas correspondientes a los hogares de los operadores de salud.

*Tabla 31 Información de entrada de los operadores de salud en el estudio de caso
Fuente: Autor*

Operador de salud	Longitud	Latitud	Habilidad q_k
1	-74.105946	4.702898	3
2	-74.047434	4.672673	5
3	-74.148019	4.609888	2
4	-74.105962	4.58058	3

Finalmente se recopila la información de los pacientes a atender en el día de simulación. Tal información fue generada por el modelo utilizando las frecuencias de solicitudes por localidad, la georreferenciación de los pacientes y agregando los demás parámetros de interés del modelo. Lo anterior es presentado en la Tabla 32.

*Tabla 32 Información de los paciente del estudio de caso
Fuente: Autor*

Paciente	Longitud	Latitud	Momento de notificación (a_i)	Duración del servicio (s_i)	Habilidad requerida (h_i)	Prioridad (p_i)	
1	2-374	-74.06364023	4.63548052	313	16	5	3
2	3-47	-74.07013464	4.61886518	373	16	2	2
3	4-145	-74.09144002	4.54022238	390	15	3	4
4	2-489	-74.04984983	4.68279928	431	14	2	4
5	2-773	-74.05311150	4.67785366	442	17	4	3
6	5-24	-74.12086546	4.52357759	491	15	4	4
7	2-331	-74.04549528	4.67880748	498	19	3	2
8	2-491	-74.06632173	4.62980035	512	16	2	4
9	6-154	-74.13844978	4.59196463	555	15	3	4
10	8-228	-74.13722594	4.61161867	563	17	4	4
11	10-1605	-74.12997883	4.70322883	613	14	2	4

ESTUDIO DE CASO

12	13-107	-74.09986519	4.63681046	625	17	5	2
13	20-17	-75.23062279	4.43985510	674	19	4	3
14	1-2137	-74.02815180	4.74165221	738	17	2	3
15	2-419	-74.06349468	4.64564586	750	19	5	2
16	1-1276	-74.03524090	4.72976405	792	17	2	4
17	4-118	-74.10012339	4.55792398	866	14	2	2
18	4-133	-74.09251931	4.54564311	913	17	5	2
19	2-722	-74.06315153	4.65096451	927	18	5	4
20	3-363	-74.07386862	4.60569016	976	18	3	2
21	3-133	-74.06693407	4.60817475	991	16	2	3
22	10-2416	-74.09052289	4.68817363	1035	11	3	2
23	13-279	-74.07379940	4.64218470	1046	14	2	4
24	6-101	-74.13872558	4.57589479	1096	14	5	4
25	18-24	-74.11854376	4.58103410	1102	18	3	2
26	13-149	-74.11240510	4.58858662	1159	11	4	3
27	8-1644	-74.17776256	4.63216794	1174	18	2	2
28	1-1062	-74.03474148	4.73317923	1215	18	3	3
29	6-18	-74.12704134	4.56310659	1234	15	5	2
30	13-757	-74.10700203	4.64234094	2678	10	5	2
31	10-2426	-74.10832462	4.66813102	1333	19	2	2

7.4. RESULTADOS DEL ESTUDIO DE CASO

En la Tabla 33 se muestran los resultados del estudio de caso variando los parámetros α y β del modelo matemático. Por otra parte las rutas que deberá seguir cada operador de salud con el fin de atender a los pacientes que le fueron asignados durante el horizonte de planeación utilizando un $\alpha=0.8$ y $\beta=0.2$ se presentan en la Tabla 34.

Tabla 33 Resultados estudio de caso
Fuente: Autor

α	β	FO Acumulada	FO_1 Acumulada	FO_2 Acumulada
0.8	0.2	469.370	377.550	91.818
0.6	0.4	498.531	320.152	178.379
0.4	0.6	512.651	210.561	302.09
0.2	0.8	542.561	156.516	386.045

MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA

Tabla 34 Rutas para los operadores de salud estudio de caso $\alpha=0.8$ $\beta=0.2$

Fuente: Autor

Operador de salud 1		Operador de salud 2		Operador de salud 3		Operador de salud 4	
H1	10-1605	H2	2-374	H3	13-149	H4	4-145
10-1605	10-2416	2-374	3-47	13-149	8-1644	4-145	4-118
10-2416	1-1062	3-47	2-773	8-1644	H3	4-118	H4
1-1062	H1	2-773	2-489	-	-	-	-
-	-	2-489	2-331	-	-	-	-
-	-	2-331	2-491	-	-	-	-
-	-	2-491	5-24	-	-	-	-
-	-	5-24	6-154	-	-	-	-
-	-	6-154	8-228	-	-	-	-
-	-	8-228	13-107	-	-	-	-
-	-	13-107	20-17	-	-	-	-
-	-	20-17	2-419	-	-	-	-
-	-	2-419	1-2137	-	-	-	-
-	-	1-2137	4-133	-	-	-	-
-	-	4-133	2-722	-	-	-	-
-	-	2-722	3-363	-	-	-	-
-	-	3-363	3-133	-	-	-	-
-	-	3-133	13-279	-	-	-	-
-	-	13-279	18-24	-	-	-	-
-	-	18-24	6-101	-	-	-	-
-	-	6-101	6-18	-	-	-	-
-	-	6-18	13.757	-	-	-	-
-	-	13.757	10-2426	-	-	-	-
-	-	10-2426	H2	-	-	-	-

Del anterior esquema de ruteo se puede observar que el operador de salud 2 es el que realiza la mayoría de las visitas. Lo anterior es ocasionado porque gran parte de los pacientes que solicitan el servicio se encuentran en la localidad de Usaquén que es el lugar de vivienda del operador, además de que se trabaja un $\alpha=0.8$ y $\beta=0.2$ que le da mayor importancia al ruteo ocasiona la gran carga laboral de éste operado de salud y finalmente es ocasionado por que es el operador de salud más calificado. Los otros operadores de salud son utilizados sólo cuando las solicitudes surgen muy cerca de sus ubicaciones y no requieren de habilidades superiores para realizar la atención. Por otra parte en los resultados de las funciones objetivo de la Tabla 33 se observa al igual que en resultados anteriores que cuando se aumenta el parámetro β (disminuye α) los valores de la función objetivo general del sistema aumentan lo que indica que la promesa de servicio a los pacientes acarrea costos mayores en el sistema que la función de ruteo.

7.5. CONCLUSIONES DEL ESTUDIO DE CASO

En este estudio de caso se mostró como ingresar información de entrada real al modelo de simulación basado en agentes del cuidado a la salud domiciliaria. Se realizó el manejo y tratamiento de las coordenadas geográficas de las ubicaciones de los pacientes hasta obtener los tiempos de desplazamiento entre los nodos, además de que se realizó el análisis de entrada de la información disponible. Se concluye que el modelo de simulación basado en agentes y su implementación computacional es lo suficientemente flexible para realizar modificaciones que permitan tratar casos de la vida real. Finalmente se presenta información de un día de simulación obteniendo los valores de las funciones objetivo y un posible esquema de ruteo para el sistema.

CONCLUSIONES, TRABAJO FUTURO Y VALIDACIÓN DE LOS OBJETIVOS DE INVESTIGACIÓN.

El cuidado a la salud domiciliaria es una alternativa a la prestación tradicional del servicio de salud que consiste en enviar profesionales médicos a los hogares de los pacientes con el fin de atender sus necesidades en materia de salud. El envío de profesionales médicos u operadores de salud implica la realización de esquemas de ruteo en los cuales se deben considerar distintos criterios de optimización así como la naturaleza dinámica del cuidado domiciliario. Dentro de los criterios de optimización se encuentran las preferencias de los operadores de salud así como de los pacientes, el punto de partida y finalización de las rutas, el tiempo de trabajo de los operadores de salud entre otros. Por otra parte la naturaleza dinámica del cuidado domiciliario yace en el ingreso de nuevas demandas cuando los operadores de salud estén realizando sus rutas lo que exige realizar nuevos esquemas de ruteo para servir de la mejor forma a éstos nuevos pacientes.

En este trabajo inicialmente se realizó una revisión del estado del arte en asignación y ruteo de operadores de salud en el cuidado domiciliario identificando los objetivos, restricciones, datos de prueba, métodos de solución, entre otros. Se observó a través de la revisión literaria que la mayoría de trabajos apuntan a resolver modelos estáticos y determinísticos en los cuales se resuelve el problema una sola vez para un horizonte de planeación. Por otra parte se encontraron los objetivos, restricciones y metodologías más usadas a la hora de resolver el problema. Se estableció que el problema de cuidado a la salud domiciliaria es de naturaleza dinámica por lo cual se debe utilizar una metodología de solución la cual permita afrontar el dinamismo del sistema. Para realizar lo anterior se escogió la simulación basada en agentes por lo que se realizó una conceptualización y una revisión de aplicaciones y plataformas para sistemas multi-agente.

Posteriormente en este trabajo se propuso un modelo de programación lineal entera mixta para resolver el problema de ruteo de operadores de salud en el cuidado a la salud domiciliaria. El modelo matemático propuesto es multi-objetivo en donde se pretende minimizar el tiempo de transporte de los operadores de salud y la promesa de servicio a los pacientes. En otras palabras se pretende minimizar el tiempo de llegada a los hogares de los pacientes y el tiempo de transporte en el sistema. El modelo fue probado varias veces y se encontró que variando los parámetros de las funciones objetivo se pueden obtener distintos esquemas de ruteo que le den más importancia al ruteo o a la promesa de servicio.

Por otra parte se realizó la aproximación multi-agente del sistema de cuidado a la salud domiciliaria utilizando diagramas UML para mostrar las interacciones, comunicación y las actividades de los agentes en el sistema. Luego se realizó la implementación del sistema multi-agente a través de la construcción del código computacional el cual fue realizado en Java a través del Eclipse, utilizando la librería de agentes JADE y empleando el modelo matemático propuesto anteriormente. De esa manera se obtuvo el modelo de simulación basado en agentes para el cuidado a la salud domiciliaria.

El modelo de simulación fue probado varias veces con el fin de establecer una configuración que lograra un balance entre los parámetros α y β del modelo matemático, la duración y el número de periodos de simulación y finalmente el número de operadores de salud necesarios para atender a 40 pacientes en un horizonte de planeación de 600 minutos. Se demostró que utilizando un $\alpha=0.4$ y $\beta=0.6$, 8 periodos de simulación de 75 minutos cada uno y 4 operadores de salud se consigue un buen tiempo promedio de espera en el sistema a un costo aceptable para el proveedor del servicio de cuidado a la salud domiciliaria. Finalmente se realizó un estudio de caso en el cual el objetivo era demostrar que el modelo de simulación basado en agentes se puede configurar para recibir información real. En el caso de estudio se manejó exitosamente la información brindada por el proveedor del servicio generando un posible esquema de ruteo para un día de operación del sistema.

Trabajos futuros apuntan a la construcción de procedimientos heurísticos que puedan ser utilizados dentro del modelo de simulación con el fin de reducir tiempos de computación debido a que el problema de ruteo es de naturaleza Np-duro. Por otra parte se apunta a la utilización de más opciones de la librería de Jade con el fin de mejorar el modelo de simulación basado en agentes del sistema de cuidado domiciliario. En el desarrollo del estudio de caso se utilizaron distancias esféricas lo que no corresponde a la realidad y puede llevar a errores en la toma de decisiones en el sistema, se sugiere utilizar información de la malla vial de Bogotá para realizar experimentaciones y elaborar escenarios que permitan mejorar la toma de decisiones del proveedor de cuidado domiciliario.

Finalmente se realiza la validación de los objetivos de investigación los cuales fueron cumplidos a cabalidad como se muestra en la Tabla 35. En la misma tabla se especifican las actividades de cada objetivo específico, el porcentaje de avance y los resultados obtenidos durante el desarrollo del proyecto. Es importante resaltar que en el transcurso del proyecto de investigación se elaboraron tres artículos científicos y se obtuvieron ponencias en eventos académicos.

Tabla 35 Validación de los objetivos del proyecto de investigación
Fuente: Autor

Objetivo específico	Actividades	Avance	Resultados
Definir las características teóricas y técnicas del problema de asignación y ruteo en el cuidado de salud en casa.	Recolección y consulta de literatura sobre asignación y ruteo de operadores de salud.	100%	Repositorio de documentos científicos sobre asignación y ruteo de operadores de salud
	Clasificación de la literatura revisada.	100%	Matriz de resúmenes de documentos. Matriz de clasificación de documentos.
	Construcción de un documento de revisión de la literatura.	100%	Artículo de revisión bibliográfica: Asignación y ruteo en el cuidado de la salud domiciliaria: revisión de la literatura. Sometido a revisión a la revista Estudios Gerenciales (Anexo 6)

MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA

Establecer los componentes y las interacciones de un modelo para la asignación y ruteo de operadores de salud.	Análisis del problema de asignación y ruteo de operadores de salud.	100%	Elementos, variables, objetivos, parámetros del sistema de cuidado a la salud domiciliaria
	Construcción de un modelo de programación matemática para solucionar el problema.	100%	Formulación del modelo de programación matemática para asignación y ruteo de operadores de salud
	Ejecución y experimentación del modelo de programación matemática.	100%	Código del modelo en XPRESS y resultados de experimentación con instancias disponibles en la literatura
	Construcción de un documento con el modelo y los resultados de su experimentación.	100%	Artículo: Modelo de programación entera mixta para programación y ruteo en cuidado a la salud domiciliaria considerando la promesa de servicio, enviado y aceptado al congreso CIO 2016 para realizar ponencia. (Anexo 7)
Implementar el modelo de simulación basada en agentes en un software especializado para ello.	Estructuración del modelo de simulación basada en agentes.	100%	Diagramas de caso de uso, diagramas de clase, diagrama de actividades, diagramas de secuencia y diagrama de árbol de agentes.
	Construcción del modelo de simulación basada en agentes.	100%	Estructura del modelo de simulación y diagrama de despliegue del mismo.
	Implementación del modelo en software específico Jade-Java.	100%	Clases, métodos, comportamientos, implementados en Java utilizando Jade.
	Pruebas de ejecución del modelo de agentes.	100%	Artículo: Multi-agent approach for solving the dynamic home health care routing problem, enviado y aceptado al congreso WEA 2016 para realizar presentación (Anexo 8)
Evaluar el desempeño del modelo de simulación basado en agentes utilizando un estudio de caso del cuidado de salud domiciliaria.	Análisis de entrada y parametrización de información.	100%	Distribuciones de entrada de datos y demás parámetros del modelo en Java.
	Experimentación y análisis de resultados.	100%	Resultados de experimentación y conclusiones
TOTAL	Estado del proyecto de investigación	100%	

REFERENCIAS

- [1] S. Yalcindag, A. Matta, and E. Sahin, "Operator assignment and routing problems in home health care services," *2012 IEEE Int. Conf. Autom. Sci. Eng.*, pp. 329–334, 2012.
- [2] N. Genet, W. Boerma, M. Kroneman, A. Hutchinson, and R. B. Saltman, "Home Care across Europe - Current structure and future challenges," *World Heal. Organ.*, p. 156, 2012.
- [3] Banco Mundial, "Tasa de crecimiento de la población," 2011. [Online]. Available: <http://www.worldbank.org/depweb/spanish/modules/social/pgr/>.
- [4] Organización Mundial de la Salud, "10 datos sobre el envejecimiento y la salud," 2015. [Online]. Available: http://www.who.int/features/factfiles/ageing/ageing_facts/es/.
- [5] A. A. Cire and J. N. Hooker, "A Heuristic Logic-Based Benders Method for the Home Health Care Problem A Heuristic Logic-Based Benders Method for the Home Health Care Problem," pp. 1–12, 2012.
- [6] G. Hiermann, M. Prandtstetter, A. Rendl, J. Puchinger, and G. R. Raidl, "Metaheuristics for solving a multimodal home-healthcare scheduling problem," *Cent. Eur. J. Oper. Res.*, pp. 1–25, 2013.
- [7] R. M. M. Cotta, M. M. Suárez-Varela, A. Llopis González, J. S. Cotta Filho, E. R. Real, and J. A. D. Ricós, "La hospitalización domiciliaria: antecedentes, situación actual y perspectivas," *Rev. Panam. Salud Pública*, vol. 10, no. 1, pp. 45–55, 2001.
- [8] E. V. Gutierrez, O. D. Galvis, D. a. López, J. S. Mock-Kow, I. Zapata, and C. J. Vidal, "Gestión Logística en la Prestación de Servicios de Hospitalización Domiciliaria en el Valle del Cauca: Caracterización y Diagnóstico," *Estud. Gerenciales*, vol. In press, no. 133, p. Under Review, 2013.
- [9] M. Mutingi and C. Mbohwa, "Home healthcare staff scheduling: A taxonomic state-of-the-Art review," *IEEE Int. Conf. Ind. Eng. Eng. Manag.*, no. iii, pp. 1107–1111, 2014.
- [10] S. Yalcindag, A. Matta, and E. Sahin, "Human Resource Scheduling and Routing Problems in Home Health Care Context: A Literature Review," pp. 1–34, 2012.
- [11] M. M. García, "Estudio Del Triaje En Un Servicio De Urgencias Hospitalario," *RevistaEnfermeríaCyL*, vol. 5, no. 1, pp. 42–49, 2013.
- [12] S. . Begur, D. M. Miller, and J. . Weaber, "An integrated spatial decision support system for scheduling and routing home health care nurses.," *Inst. Oper. Res. Manag. Sci.*, pp. 35–48, 1997.
- [13] E. Cheng and J. L. Ricks, "A Home Health Care Routing and Scheduling Problem," 1998.
- [14] Z. Yuan and A. Fügenschuh, "Home Health Care Scheduling: A Case Study," vol. 22, 2015.
- [15] Y. Kergosien, C. Lenté, and J. Billaut, "Home health care problem An extended multiple Traveling Salesman Problem," *Multidiscip. Int. Conf. Sched. Theory Appl. (MISTA 2009)*, no. August, pp. 1–8, 2009.
- [16] R. Ben Bachouch, A. Guinet, and S. Hajri-gabouj, "An optimization model for task assignment in home health care," 2010.

- [17] V. Borsani, A. Matta, G. Beschi, and F. Sommaruga, "A Home Care Scheduling Model For Human Resources," pp. 0–5, 2006.
- [18] B. Issaoui, I. Zidi, E. Marcon, and K. Ghedira, "New multi-objective approach for the home care service problem based on scheduling algorithms and variable neighborhood descent," *Electron. Notes Discret. Math.*, vol. 47, pp. 181–188, 2015.
- [19] S. Yalcindag, P. Cappanera, M. Grazia Scutellà, E. Sahin, and A. Matta, "Pattern-based decompositions for human resource planning in home health care services," *Comput. Oper. Res.*, vol. 73, pp. 12–26, 2016.
- [20] P. Cappanera and M. G. Scutellà, "Home care optimization: Impact of pattern generation policies on scheduling and routing decisions," *Electron. Notes Discret. Math.*, vol. 41, pp. 53–60, 2013.
- [21] K. Braekers, R. F. Hartl, S. N. Parragh, and F. Tricoire, "A bi-objective home care scheduling problem: Analyzing the trade-off between costs and client inconvenience," *Eur. J. Oper. Res.*, vol. 248, no. 2, pp. 428–443, 2016.
- [22] B. Yuan, R. Liu, and Z. Jiang, "Home Health Care Crew Scheduling and Routing Problem with Stochastic Service Times," in *Automation Science and Engineering (CASE), 2014 IEEE International Conference on*, 2014, p. 564,569.
- [23] A. F. Torres Ramos, C. L. D. Quintero Araujo, and E. H. D. Alfonso Lizarazo, "Optimización de las operaciones de programación y ruteo de personal especializado en servicios médicos domiciliarios," Universidad de La Sabana, 2015.
- [24] G. Stiglic and P. Kokol, "Intelligent patient and nurse scheduling in ambulatory health care centers," *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, vol. 5, pp. 5475–5478, 2005.
- [25] R. Redjem, E. Marcon, and X. Xie, "Management of the care activities in Home Health Care Services : The routing and scheduling of caregivers level," pp. 712–718, 2014.
- [26] J. F. Bard, Y. Shao, X. Qi, and A. I. Jarrah, "The traveling therapist scheduling problem," *IIE Trans.*, vol. 46, no. 7, pp. 683–706, 2014.
- [27] C. Akjiratikarl, P. Yenradee, and P. R. Drake, "PSO-based algorithm for home care worker scheduling in the UK," *Comput. Ind. Eng.*, vol. 53, no. 4, pp. 559–583, 2007.
- [28] J. Steeg and M. Schröder, "A Hybrid Approach to Solve the Periodic Home Health Care Problem," *Oper. Res. Proc. 2007 Sel. Pap. Annu. Int. Conf. Ger. Oper. Res. Soc. Saarbrücken, Sept. 5–7, 2007*, pp. 297–302, 2008.
- [29] L. En-nahli, H. Allaoui, and I. Nouaouri, "A Multi-objective Modelling to Human Resource Assignment and Routing Problem for Home Health Care Services," *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 698–703, 2015.
- [30] P. a. Maya Duque, M. Castro, K. Sorensen, and P. Goos, "Home care service planning. the case of Landelijke Thuiszorg," *Eur. J. Oper. Res.*, vol. 243, no. 1, pp. 292–301, 2015.
- [31] S. Nickel, M. Schröder, and J. Steeg, "Mid-term and short-term planning support for home health care services," *Eur. J. Oper. Res.*, vol. 219, no. 3, pp. 574–587, 2012.
- [32] G. Carello and E. Lanzarone, "A cardinality-constrained robust model for the assignment problem in Home Care services," *Eur. J. Oper. Res.*, vol. 236, no. 2, pp. 748–762, 2014.

- [33] P. Cappanera and M. G. Scutellà, "Joint Assignment, Scheduling, and Routing Models to Home Care Optimization: A Pattern-Based Approach," *Transportation Sci.*, no. December, 2014.
- [34] H. Allaoua, S. Borne, L. Létocart, and R. Wolfler Calvo, "A matheuristic approach for solving a home health care problem," *Electron. Notes Discret. Math.*, vol. 41, pp. 471–478, 2013.
- [35] M. Masmoudi and K. Mellouli, "MILP for Synchronized-mTSPTW : application to Home HealthCare Scheduling," pp. 297–302, 2014.
- [36] R. Redjem and E. Marcon, "Operations management in the home care services: a heuristic for the caregivers??? routing problem," *Flex. Serv. Manuf. J.*, pp. 280–303, 2015.
- [37] B. Elbenani, J. a. Ferland, and V. Gascon, "Mathematical programming approach for routing home care nurses," *2008 IEEE Int. Conf. Ind. Eng. Eng. Manag.*, pp. 107–111, 2008.
- [38] M. S. Rasmussen, T. Justesen, A. Dohn, and J. Larsen, "The Home Care Crew Scheduling Problem : Preference-based visit clustering and temporal dependencies," *Eur. J. Oper. Res.*, vol. 219, no. 3, pp. 598–610, 2012.
- [39] R. Redjem, S. Kharraja, X. Xie, and E. Marcon, "Coordinated Multi - criteria Scheduling of Caregivers in Home Health Care Services," pp. 519–524, 2011.
- [40] M. Mutingi and C. Mbohwa, "A fuzzy simulated evolution algorithm for multi-objective homecare worker scheduling," *IEEE Int. Conf. Ind. Eng. Eng. Manag.*, pp. 586–590, 2013.
- [41] K.-D. Rest and P. Hirsch, "Daily scheduling of home health care services using time-dependent public transport," *Flex. Serv. Manuf. J.*, 2015.
- [42] A. Trautsamwieser and P. Hirsch, "Optimization of daily scheduling for home health care services," *J. Appl. Oper. Res.*, vol. 3, no. 3, pp. 124–136, 2011.
- [43] S. Chahed, E. Marcon, E. Sahin, D. Feillet, and Y. Dallery, "Exploring new operational research opportunities within the Home Care context: The chemotherapy at home," *Health Care Manag. Sci.*, vol. 12, no. 2, pp. 179–191, 2009.
- [44] F. Gayraud, L. Deroussi, N. Grangeon, and S. Norre, "A New Mathematical Formulation for the Home Health Care Problem," *Procedia Technol.*, vol. 9, pp. 1041–1047, 2013.
- [45] S. R. Ait Haddadene, N. Labadie, and P. Caroline, "GRASP for the Vehicle Routing Problem with Time Windows, synchronization and precedence constraints," (*eHPWAS'14*), vol. 7, pp. 10–23, 2014.
- [46] D. Bredström and M. Rönnqvist, "Combined vehicle routing and scheduling with temporal precedence and synchronization constraints," *Eur. J. Oper. Res.*, vol. 191, no. 1, pp. 19–29, 2008.
- [47] A. Torres-Ramos, E. Alfonso-Lizarazo, L. Reyes-Rubiano, and C. Quintero-Araújo, "Mathematical Model for the Home Health Care Routing and Scheduling Problem with Multiple Treatments and Time Windows," *Proc. 1st Int. Conf. Math. Methods Comput. Tech. Sci. Eng. (MMCTSE 2014)*, no. JANUARY, pp. 140–145, 2014.
- [48] J. F. Bard, Y. Shao, and H. Wang, "Weekly scheduling models for traveling therapists," *Socioecon. Plann. Sci.*, vol. 47, no. 3, pp. 191–204, 2013.

- [49] N. Triki, T. Garaix, and X. Xie, "A two-phase approach for periodic home health care planning," pp. 518–523, 2014.
- [50] F. Rothlauf, "Design of Modern Heuristics: Principles and Application," Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 45–102.
- [51] P. Brucker, R. Qu, and E. Burke, "Personnel scheduling: Models and complexity," *Eur. J. Oper. Res.*, vol. 210, no. 3, pp. 467–473, 2011.
- [52] M. Liu, L. Xu, and Q. Su, "Mathematical Model for Pattern-based Medical Team Planning Problem," 2014.
- [53] S. Errarhout, Afrae Hkarraja and I. Zorkan, "Caregivers' Assignment Problem in Home Health Care Structures Home Care Optimization: A Pattern-Based Approach," no. October, 2013.
- [54] D. Aiane, A. El-Amaraoui, and K. Mesghouni, "A New Optimization Approach for a Home Health Care Problem," no. October, 2015.
- [55] R. Liu, X. Xie, V. Augusto, and C. Rodriguez, "Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care," *Eur. J. Oper. Res.*, vol. 230, no. 3, pp. 475–486, 2013.
- [56] F. Luna, a Cervantes, and P. Isasi, "Large-Scale Home Care Crew Scheduling with a Parallel Evolutionary Algorithm," *P2P, Parallel, Grid, Cloud Internet Comput. (3PGCIC), 2013 Eighth Int. Conf.*, pp. 588–593, 2013.
- [57] M. Mısır, K. Verbeeck, P. De Causmaecker, and G. Vanden Berghe, "Hyper-heuristics with a Dynamic Heuristic Set for the Home Care Scheduling Problem," 2010.
- [58] J. Jemai, M. Chaieb, and K. Mellouli, "The Home Care Scheduling Problem : A modeling and solving issue," 2013.
- [59] A. Hertz and N. Lahrichi, "Client assignment algorithms for home care services," no. June 2004, pp. 1–28, 2006.
- [60] R. Liu, X. Xie, and T. Garaix, "Weekly home health care logistics," pp. 282–287, 2013.
- [61] B. Tozlu, R. Daldal, T. Ünlüyurt, and B. Çatay, "Crew Constrained Home Care Routing Problem with Time Windows," 2015.
- [62] Z. Dan, L. Cai, and L. Zheng, "Improved Multi-Agent System for the Vehicle Routing Problem with Time Windows," *Tsinghua Sci. Technol.*, vol. 14, no. 3, pp. 407–412, 2009.
- [63] Y. Chen, X. Wang, F. Lin, and K. Tang, "The multi-agent modelling of production logistics system and the shop dynamic scheduling optimization in discrete manufacturing enterprises," *Technol. Innov. Conf. 2006. ITIC 2006. Int.*, pp. 604–610, 2006.
- [64] J. M. Usher, "Negotiation-based routing in job shops via collaborative agents," *J. Intell. Manuf.*, vol. 14, no. 5, pp. 485–499, 2003.
- [65] R. Erol, C. Sahin, A. Baykasoglu, and V. Kaplanoglu, "A multi-agent based approach to dynamic scheduling of machines and automated guided vehicles in manufacturing systems," *Appl. Soft Comput. J.*, vol. 12, no. 6, pp. 1720–1732, 2012.
- [66] M. Mutingi and C. Mbohwa, "A Fuzzy-Based Particle Swarm Optimization Approach for Task Assignment in Home Health Care," *South African J. Ind. Eng.*, vol. 25, no.

- November, pp. 84–95, 2014.
- [67] P. M. Koeleman, S. Bhulai, and M. Van Meersbergen, “Optimal patient and personnel scheduling policies for care-at-home service facilities,” *Eur. J. Oper. Res.*, vol. 219, no. 3, pp. 557–563, 2012.
- [68] L. Álvarez, C. Caicedo, M. Malaver, and G. Méndez, “Investigación y desarrollo de un prototipo de Sistema Experto para Scheduling en Pymes con entorno Job Shop,” *Rev. ...*, no. 12, pp. 125–136, 2011.
- [69] K. Metaxiotis, D. Askounis, and J. Psarras, “Expert systems in production planning and scheduling: A state-of-the-art survey,” *J. Intell. Manuf.*, 2002.
- [70] G. Méndez, L. Álvarez, C. Caicedo, and M. Malaver, *Sistema Experto para la programación de producción-investigación y desarrollo de un prototipo.*, Primera. Bogotá, 2013.
- [71] J. Wirnitzer, I. Heckmann, A. Meyer, S. Nickel, T. Wirnitzer, D. Sris Re[™]km—nn, D. Enne Weyer, and D. ftef—n Xi[™]kel, “Patient-based nurse rostering in home care,” *Oper. Res. Heal. Care*, vol. 8, pp. 91–102, 2015.
- [72] R. J. Acosta-amado, I. Atuahene, and R. Sawhney, “An Integer Programming Model for Home Visitation (HV) Deployment,” *Proc. 2013 Ind. Syst. Eng. Res. Conf.*, no. 1997, pp. 1442–1449, 2013.
- [73] M. Pinedo, C. Zacharias, and N. Zhu, “Scheduling in the service industries: An overview,” *J. Syst. Sci. Syst. Eng.*, vol. 3756, 2015.
- [74] L. Talarico, F. Meisel, and K. Sörensen, “Ambulance routing for disaster response with patient groups,” *Comput. Oper. Res.*, vol. 56, pp. 120–133, 2015.
- [75] L. Zhen, S. Sheng, Z. Xie, and K. Wang, “Decision rules for ambulance scheduling decision support systems,” *Appl. Soft Comput. J.*, vol. 26, pp. 350–356, 2014.
- [76] P. Eveborn, P. Flisberg, and M. Rönnqvist, “LAPS CARE — an operational system for staff planning of home care,” vol. 171, pp. 962–976, 2006.
- [77] S. Bertels and T. Fahle, “A hybrid setup for a hybrid scenario: Combining heuristics for the home health care problem,” *Comput. Oper. Res.*, vol. 33, no. 10, pp. 2866–2890, 2006.
- [78] B. Bashir, M. Chabrol, and C. Caux, “Literature review in home care,” *9th Int. Conf. Model. Optim. Simul.*, 2012.
- [79] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. 1995.
- [80] M. Wooldige, “An Introduction to Multiagent Systems,” 2009.
- [81] C. M. Macal and M. J. North, “Teaching Agent-Based Modeling and Simulation,” *Mil. Oper. Res.*, pp. 268–277, 2010.
- [82] K. Rajan, M. Shirley, W. Taylor, and B. Kanefsky, “Ground tools for autonomy in the 21st century,” *2000 IEEE Aerosp. Conf. Proc. (Cat. No.00TH8484)*, vol. 7, 2000.
- [83] C. M. Macal and M. J. North, “INTRODUCTORY TUTORIAL: AGENT-BASED MODELING AND SIMULATION,” pp. 1451–1464, 2011.
- [84] P. Davidsson, “Multi agent based simulation: Beyond social simulation,” *Lect. Note. Artif. Intell.*, vol. 1979, pp. 97–107, 2001.
- [85] H. V. D. Parunak, R. Savit, and R. L. Riolo, “Agent-Based Modeling vs . Equation-

- Based Modeling : A Case Study and Users ' Guide," in *Multi-Agent Systems and Agent-Based Simulation*, 1998, pp. 1–16.
- [86] R. G. Sargent, "AGENT-BASED SIMULATION TUTORIAL - SIMULATION OF EMERGENT BEHAVIOR AND DIFFERENCES BETWEEN AGENT-BASED SIMULATION AND DISCRETE-EVENT SIMULATION," *Simulation*, no. 2001, pp. 135–150, 2010.
- [87] C. M. Macal and M. J. North, "TUTORIAL ON AGENT-BASED MODELING AND SIMULATION PART 2: HOW TO MODEL WITH AGENTS," pp. 73–83, 2006.
- [88] W. C. Regli, S. Member, I. Mayk, S. Member, C. T. Cannon, J. B. Kopena, R. N. Lass, S. Member, W. M. Mongan, D. N. Nguyen, J. K. Salvage, E. A. Sultanik, and K. Usbeck, "Development and Specification of a Reference Architecture for Agent-Based Systems," vol. 44, no. 2, pp. 146–161, 2014.
- [89] J. Moreno, "MODELO DE SIMULACIÓN BASADO EN AGENTES DE SOFTWARE INTELIGENTES PARA EL APRENDIZAJE DE ESTRATEGIAS DE COMERCIALIZACIÓN DE ENERGÍA ELÉCTRICA EN COLOMBIA," Universidad Nacional de Colombia, 2007.
- [90] J. J. Gomez, "Metodologías para el desarrollo de sistemas multi-agente," *Rev. Iberoam. Intel. Artif.*, vol. 18, no. 18, pp. 51–63, 2003.
- [91] R. H. Bordini, L. Braubach, J. J. Gomez-sanz, G. O. Hare, A. Pokahr, and A. Ricci, "A Survey of Programming Languages and Platforms for Multi-Agent Systems," *Informatica*, vol. 30, no. 1, pp. 33–44, 2006.
- [92] K. Kravari and N. Bassiliades, "A Survey of Agent Platforms," *J. Artif. Soc. Soc. Simul.*, vol. 18, no. 1, p. 11, 2015.
- [93] Z. Yu-xian, L. Lei, W. Hong, Z. Yan-yan, G. Xu, and C. Meng, "Approach to the distributed job shop scheduling based on multi-agent," *2008 IEEE Int. Conf. Autom. Logist.*, no. September, pp. 2031–2034, 2008.
- [94] D. Barbucha and P. Jędrzejowicz, "Multi-agent platform for solving the dynamic vehicle routing problem," *Intell. Transp. Syst. 2008. ITSC 2008. 11th Int. IEEE Conf.*, pp. 517–522, 2008.
- [95] S. Abras, C. Kieny, S. Ploix, and F. Wurtz, "MAS architecture for energy management: Developing smart networks with JADE platform," *2013 IEEE Int. Conf. Smart Instrumentation, Meas. Appl. ICSIMA 2013*, no. November, pp. 26–27, 2013.
- [96] L. Fasanotti, "A distributed intelligent maintenance system based on artificial immune approach and multi-agent systems," *Proc. - 2014 12th IEEE Int. Conf. Ind. Informatics, INDIN 2014*, pp. 783–786, 2014.
- [97] M. Xu, "Production scheduling of agile manufacturing based on multi-agents," *2009 2nd Int. Symp. Knowl. Acquis. Model. KAM 2009*, vol. 2, pp. 323–325, 2009.
- [98] A. M. Wahaishi and R. O. Aburukba, "An Agent-Based Personal Assistant for Exam Scheduling," in *Computer and Information Technology (WCCIT)*, 2013.
- [99] A. E. Garcés, J. Moreno, and S. Múnera, "Modelo de simulación de una subasta de doble punta mediante el paradigma multi-agente.," *Rev. Av. en Sist. e Informática*, vol. 6, no. 1, pp. 197–206, 2009.
- [100] L. A. De Santa-Eulalia, S. D'Amours, and J. M. Frayret, "Modeling agent-based simulations for supply chain planning: The FAMASS methodological framework,"

- Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern.*, pp. 1710–1718, 2010.
- [101] Y. Zhang, Q. Wu, X. Tang, and Z. Ma, “Progresses for multi-agents system models,” *2010 2nd Int. Conf. Comput. Autom. Eng. ICCAE 2010*, vol. 3, pp. 44–48, 2010.
- [102] P. Davidsson, P. Davidsson, J. Holmgren, J. Holmgren, H. Kyhlbäck, H. Kyhlbäck, D. Mengistu, and D. Mengistu, “Applications of Agent Based Simulation,” *Simulation*, pp. 15–27, 2007.
- [103] Networking and Emerging Optimization Research Group - Universidad de Málaga, “Multiple Depot VRP with Time Windows Instances.” [Online]. Available: <http://neo.lcc.uma.es/vrp/vrp-instances/multiple-depot-vrp-with-time-windows-instances/>.
- [104] J. Rumbaugh, I. Jacobson, and G. Booch, *The UML reference manual*, vol. 1. 1999.
- [105] Universidad Nacional de Colombia Sede Amazonía, “Tips y Trucos de Gis.” [Online]. Available: <http://www.unal.edu.co/siamac/sig/tips.html>.
- [106] Secretaría Distrital de Movilidad, “MOVILIDAD EN CIFRAS 2014.”
- [107] L. E. Ramirez Rodriguez, “Una solución al problema de ruteo de vehículos abierto (OVRP), implementando la heurística del vecino más cercano,” Universidad Distrital Francisco José de Caldas, 2016.

ANEXOS

Anexo 1: clasificación de la revisión bibliográfica consolidada

En la Tabla 37 se presenta de forma tabulada y consolidada el resumen de los artículos revisados correspondientes a los modelos de asignación y ruteo de operadores de salud en el cuidado domiciliario. Se seguirá la nomenclatura encontrada en la Tabla 36

Tabla 36 Nomenclatura de la consolidación de la revisión bibliográfica
Fuente: Autor

Nomenclatura		Descripción
Comp.	A	Modelo de asignación
	A Y R	Modelo de asignación y ruteo
Tipo	DET	Modelo determinístico
	EST	Modelo estocástico
Objetivo	T	Minimización de tiempo de viaje o espera
	C	Minimización de costos operacionales
	D	Minimización de la distancia recorrida por los operadores de salud
	BT	Maximización del balance de trabajo entre los operadores de salud
	S	Maximización de la satisfacción de los pacientes
	VR	Minimización de las violaciones de las restricciones
	#O	Minimización del número de operadores de salud
	M	Múltiples objetivos
Restricciones	VT	Ventanas de tiempo de los pacientes
	H	Habilidades de los operadores de salud
	P	Preferencias de los pacientes y/o operadores de salud
	TT	Tiempo de trabajo de los operadores de salud
	PR	Precedencia de las visitas domiciliarias
	S	Sincronización o coordinación de operadores de salud en una visita
	PPF	Punto de partida y finalización de los operadores de salud (UD = Un depósito, MD = Múltiples depósitos, A = Ambos casos considerados, NE = No especificado)
Metod.	EX	Métodos Exactos
	AP	Métodos Aproximados
	OT	Otros Métodos
Datos	RE	Datos Reales
	GE	Datos Generadas

Tabla 37 Clasificación consolidada de las referencias revisadas
Fuente: Autor

Ref.	Comp.		Tipo		Objetivo								Restricciones							Metod.			Datos	
	A	AYR	DET	EST	T	C	D	BT	S	VR	#O	M	VT	H	P	TT	PR	S	PPF	EX	AP	OT	RE	GE
1		✓		✓			✓							✓	✓	✓			UD	✓			✓	
5		✓	✓			✓							✓	✓		✓			UD		✓		✓	
6		✓	✓		✓			✓		✓		✓	✓	✓	✓				MD	✓	✓		✓	
14		✓	✓		✓	✓						✓	✓			✓			MD	✓	✓		✓	
15		✓	✓			✓							✓	✓	✓	✓	✓	✓	MD	✓				✓
16	✓		✓					✓					✓	✓		✓			NE	✓				✓
17	✓		✓			✓						✓		✓	✓	✓			NE	✓			✓	
18		✓	✓			✓			✓			✓	✓	✓		✓			NE	✓	✓			✓
19		✓	✓					✓	✓					✓	✓				UD	✓			✓	
20		✓	✓					✓						✓	✓				UD	✓			✓	
21		✓	✓			✓			✓			✓	✓			✓			UD	✓	✓			✓
22		✓		✓		✓			✓			✓	✓						UD	✓				✓
23		✓	✓	✓	✓								✓			✓			MD	✓			✓	
24	✓			✓	✓														NE			✓		✓
25		✓	✓		✓							✓	✓			✓	✓		UD	✓				✓
26		✓	✓			✓							✓	✓	✓	✓			MD	✓			✓	
27		✓	✓					✓					✓						MD		✓		✓	
28		✓	✓			✓	✓		✓			✓	✓			✓			NE		✓			✓
29		✓	✓		✓	✓		✓	✓			✓	✓	✓					UD	✓				✓
30		✓	✓					✓	✓			✓	✓			✓			MD	✓	✓		✓	
31		✓	✓					✓	✓			✓	✓	✓	✓				UD		✓		✓	
32	✓			✓		✓			✓			✓		✓	✓	✓			NE	✓			✓	
33		✓	✓			✓		✓				✓	✓	✓	✓				UD	✓	✓		✓	
34		✓	✓								✓		✓	✓	✓				MD	✓	✓			✓
35		✓	✓		✓							✓	✓					✓	UD	✓	✓			✓
36		✓	✓		✓								✓			✓	✓		UD		✓		✓	
37		✓	✓			✓							✓		✓		✓		MD	✓	✓		✓	
38		✓	✓			✓			✓			✓	✓				✓		MD	✓			✓	
39		✓	✓		✓								✓		✓				UD	✓				✓
40		✓	✓					✓		✓		✓	✓			✓		✓	UD			✓		✓
41		✓	✓		✓								✓	✓	✓				UD	✓	✓		✓	
42		✓	✓		✓				✓			✓	✓	✓	✓	✓			UD	✓	✓		✓	
44		✓	✓		✓			✓				✓	✓	✓			✓		UD	✓				✓
45		✓	✓		✓				✓			✓	✓				✓	✓	MD	✓	✓			✓
47		✓	✓		✓								✓	✓	✓				MD	✓			✓	
48		✓	✓			✓							✓			✓			A	✓	✓			✓
49		✓	✓			✓		✓					✓	✓					MD	✓	✓		✓	
52		✓	✓			✓								✓	✓		✓		UD	✓				✓
53	✓		✓					✓				✓		✓	✓		✓		NE	✓				✓
54		✓	✓		✓								✓						UD	✓			✓	
55		✓	✓			✓							✓				✓	✓	UD	✓	✓			✓

MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA

56		✓	✓					✓		✓	✓					NE	✓	✓		✓	
57		✓	✓		✓				✓		✓	✓	✓	✓		A		✓			✓
58		✓	✓				✓					✓	✓			UD	✓	✓			✓
60		✓	✓				✓			✓	✓				✓	UD		✓			✓
61		✓	✓				✓					✓	✓			UD	✓	✓			✓
66	✓			✓				✓		✓	✓	✓		✓		NE		✓	✓		
67	✓			✓		✓								✓		NE		✓	✓		✓
71		✓	✓							✓			✓	✓		NE	✓			✓	
72		✓	✓			✓				✓	✓			✓		NE	✓				✓
76		✓	✓			✓						✓	✓			NE	✓	✓		✓	
77		✓	✓		✓					✓		✓	✓	✓		NE	✓	✓			✓

Anexo 2: Código de Xpress del modelo matemático

model CUIDADO_DOMICILIARIO

uses "mmxprs","mmsystem","mmive"

parameters

inputfile="DATOSPASAR4.txt"

end-parameters

declarations

nop:integer !numero de cuidadores

VV:set of integer !todos los nodos

xx,yy,aa,ss,hh,pp:array(VV)of real!coordenadas

K:range !Operadores de salud

qq,ll,uu:array(K) of real !Parámetros auxiliares

end-declarations

initializations from inputfile !lee la informacion

nop

[xx,yy,aa,ss,hh,pp] as 'paciente'

[qq,ll,uu] as "operadores"

end-initializations

P:={nop..VV.size-1} ! construye el conjunto P

KI:={0..K.size-1} ! construye el conjunto KI

KF:={VV.size..VV.size+K.size-1} ! construye el conjunto KF

declarations

a,s,h,p:array(P)of real !Parámetros

q:array(K) of real !Parámetros

l:array(KI) of real!Parámetros

u: array(KF) of real !Parámetros

solucion:linctr

alfa:real

beta:real

end-declarations

!Parametros

forall (i in P) a(i):=aa(i)

forall (i in P) s(i):=ss(i)

```
forall (i in P) h(i):=hh(i)
forall (i in P) p(i):=pp(i)
forall (k in K) q(k):=qq(k)
forall (k in KI) l(k):=ll(k)
forall (k in KF) u(k):=uu(k-VV.size)
```

```
V:=KI+P+KF
V1:=KI+P
V2:=P+KF
GG:=getsize(P)
MM:=getsize(KI)
NN:=GG+MM
```

declarations

```
tiem:dynamic array(range,range) of real !Tiempo de viaje entre los nodos
X: array(V,V,K)of mpvar !1 Si el operador de salud k cubre la ruta i-j - 0 dlc
Y: array(V,K) of mpvar !Tiempo de llegada del operador de salud k al paciente i
Z:array(P) of mpvar ! Promesa de servicio al paciente i
```

end-declarations

!Calcula la matriz de tiempos o distancias

```
forall(i in KI,j in P)tiem(i,j):=abs(xx(j)-xx(i))+abs(yy(j)-yy(i))
forall(i in P,j in KI)tiem(i,j+VV.size):=abs(xx(j)-xx(i))+abs(yy(j)-yy(i))
forall(i in P,j in P|i<>j)tiem(i,j):=abs(xx(j)-xx(i))+abs(yy(j)-yy(i))
forall(i in P,j in P|i=j)tiem(i,j):=10000
```

```
forall(i in V,j in V|i<>j,k in K) create(X(i,j,k))
forall(i in V,j in V|i<>j,k in K) X(i,j,k) is_binary
forall(i in V,k in K) create(Y(i,k))
forall(i in P) create(Z(i))
BM:=110000
alfa:=0.8
beta:=0.2
```

```
solucion:=(alfa*sum(i in V,j in V,k in K | exists(X(i,j,k)))tiem(i,j)*X(i,j,k))+(beta*(sum(i in P)Z(i)/P.size))
ruteo:=sum(i in V,j in V,k in K | exists(X(i,j,k)))tiem(i,j)*X(i,j,k)
servicio:=sum(i in P)Z(i)/P.size
```

!1. Todas las demandas sean cubiertas

```
forall(j in P)sum(k in K,i in V1 | exists(X(i,j,k)))X(i,j,k)=1 !1. Todas las demandas sean cubiertas
```

!2. Flujo

```
forall(k in K,e in P) sum(j in V1)X(j,e,k)=sum(j in V2)X(e,j,k)
```

!3. Salgan de los depósitos propios

```
forall(k in K, i in V2|i=k) sum(j in V2)X(i,j,k)=1
```

!4. No salgan de depósitos ajenos

```
forall(k in K) sum(i in KI,j in V2|i<>k) X(i,j,k)=0
```

!5. Llegan a sus propios depósitos

```
forall(k in K) sum(i in V1)X(i,k+NN,k)=1
```

!6. No lleguen a depósitos ajenos

```
forall(k in K) sum(i in V1,j in KF|j<>k+NN)X(i,j,k)=0
```

*MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL
PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA*

!7. Cálculo de llegada a los pacientes

forall(k in K,i,j in P) $Y(i,k)+tiem(i,j)+s(i)-BM*(1-X(i,j,k))\leq Y(j,k)$

!8. Cálculo del tiempo de salida de los depósitos

forall(k in K,i in KI,j in V2) $Y(i,k)+tiem(i,j)-BM*(1-X(i,j,k))\leq Y(j,k)$

!9. Tiempo de llegada a los depósitos

forall(k in K,i in V1,j in KF) $Y(i,k)+tiem(i,j)-BM*(1-X(i,j,k))\leq Y(j,k)$

!10. Cumplimiento de la ventana de tiempo inferior

forall(k in K,i in P | exists(Y(i,k))) $Y(i,k)\geq a(i)$

!11. Función TRIAGE

forall(i in P) $(16.071*(p(i)^2)-(37.929*p(i))+24+a(i))\leq Z(i)$

!12. Relación promesa de servicio con tiempos de llegada

forall(k in K,i,j in P | exists(Y(i,k))) $Y(i,k)\leq Z(i)$

!13. Operador de salud suficientemente calificado realiza las visitas

forall(k in K,i in V1,j in P) $h(j)-BM*(1-X(i,j,k))\leq q(k)$

!14. El operador de salud inicie los tours después del inicio de su jornada laboral

forall(k in K,i in KI) $Y(i,k)\geq l(i)$

!15. El operador de salud termine los tours antes de que termine su jornada laboral

forall(k in K,j in KF) $Y(j,k)\leq u(j)$

minimize(solucion)

writeln("Función Objetivo: ",getobjval)

writeln("Ruteo: ",getact(ruteo))

writeln("Servicio: ",getact(servicio))

forall(k in K)do

writeln("Operador de salud ", k)

forall(i,j in V|i<>j) do

if (getsol(X(i,j,k))<>0)then

write(i," - ",j," ")

end-if

end-do

writeln("")

end-do

writeln("")

forall (i in KI,j in P,k in K) do

if (getsol(X(i,j,k))=1)

then

writeln("El operador de salud ",k," sale desde su casa en ",i," en el tiempo ",getsol(Y(i,k)))

end-if

end-do

writeln("")

forall (i in V1,j in P,k in K) do

if (getsol(X(i,j,k))=1 and (getsol(Y(j,k))>0))

then

writeln("El operador de salud ",k," llega al paciente ",i," en el tiempo ",getsol(Y(j,k)))

```

end-if
end-do
writeln("")

forall (i in V2,j in KF,k in K|j>NN) do
if (getsol(X(i,j,k))=1 and (getsol(Y(j,k))>0))
then
writeln("El operador de salud ",k," llega a su casa en ",j," en el tiempo ",getsol(Y(j,k)))
end-if
end-do
writeln("")
forall (i in P) do
writeln("Promesa de servio a paciente ",i," : ",getsol(Z(i)))
end-do

end-model

```

Anexo 3: Código en Java del modelo de simulación basado en Agentes

Clase AgenteCoordinador

```

import jade.core.Agent;
import jade.core.behaviours.SimpleBehaviour;
import jade.lang.acl.*;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;
import com.dashoptimization.XPRSexception;
import jade.core.AID;

public class AgenteCoordinador extends Agent{
    /***DATA***/

    double alpha=1;
    double beta=0;
    /***SETS***/
    int K;    /* Number of Operators*/
    int N;    /* Number of Nodes*/
    int P;    /* Number of Patients*/
    int KI;   /* Number of Depots*/
    /***DATA***/
    double [] Nodes ;    /* Nodes */
    double[] Depots ;    /* Nodes */
    double[] cx ;        /* x */
    double[] cy ;        /* y */
    double[] a ;        /* lower bound of time windows*/
    double[] s ;        /* service time*/
    double[] h ;        /* skills of operators*/
    double[] p ;        /* triage */
    double [] q;        /* capacity */
    double [] l;        /* lower bound of depot */
    double [] u;        /* upper bound of depot */
    double gama;        /* parameter of triage */
    double sigma;       /* parameter of triage */
    double epsilon;     /* parameter of triage */

    ArrayList<ArrayList<Integer>> rutas;
    ArrayList<Double> zz;
    ArrayList<ArrayList<Double>> pathst;
    ArrayList<Double> yy;

    public void setup(){

```

*MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL
PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA*

```

        System.out.println("Hola, soy el agente"+getLocalName());

        addBehaviour(new HacerRuteo());

    }
    protected void takeDown(){
        System.out.println("Ejecucion finalizada "+getLocalName());
    }

    private class HacerRuteo extends SimpleBehaviour{

        public void action() {
            ACLMessage aviso = receive();
            if(aviso!=null){
                System.out.println("Datos generados, Leyendo información");
                String ii=aviso.getContent();
                String[] divi = ii.split("_");
                int [] info=new int[divi.length];
                for(int b=0;b<info.length;b++){
                    info[b]=Integer.parseInt(divi[b]);
                }
                int iii=info[0];
                int period=info[1];
                String name_routing="Datos"+iii+".txt";
                LeerDatos Instance=new LeerDatos(name_routing);
                K=Instance.K;
                P=Instance.P;
                gama=Instance.gama;
                sigma=Instance.sigma;
                epsilon=Instance.epsilon;
                N=P+K+K;
                Nodes = new double[N];
                cx=new double[N];
                cy=new double[N];
                a=new double[N];
                s=new double[N];
                h=new double[N];
                p=new double[N];
                /* Nodes */
                for (int i = 0; i<=P+K-1; i++){
                    Nodes[i]=Instance.Nodes[i];
                    cx[i]=Instance.cx[i];
                    cy[i]=Instance.cy[i];
                    a[i]=Instance.a[i];
                    s[i]=Instance.s[i];
                    h[i]=Instance.h[i];
                    p[i]=Instance.p[i];
                }
                Depots=new double[K];
                q=new double[K];
                l=new double[K];
                u=new double[K];
                for (int i = 0; i<K; i++){
                    Depots[i]=Instance.Depots[i];
                    q[i]=Instance.q[i];
                    l[i]=Instance.l[i];
                    u[i]=Instance.u[i];
                }
                System.out.println("Generando ruteo de la iteración: "+(iii+1));
                try {
                    Modelomatemat ruteo=new
                    Modelomatemat(alpha,beta,K,N,P,Nodes,Depots,cx,cy,a,s,h,p,q,l,u,gama,sigma,epsilon);
                    rutas=ruteo.path;
                    zz=ruteo.zz;
                    yy=ruteo.yy;
                    pathst=ruteo.pathst;

                    ArrayList<Integer> a1;
                    ArrayList<Integer> a2;

```

```

ArrayList<Integer> a3;
ArrayList<Integer> a4;
ArrayList<Double> b1;
ArrayList<Double> b2;
ArrayList<Double> b3;
ArrayList<Double> b4;

a1=rutas.get(0);
a2=rutas.get(1);
a3=rutas.get(2);
a4=rutas.get(3);
b1=pathst.get(0);
b2=pathst.get(1);
b3=pathst.get(2);
b4=pathst.get(3);

String iii=String.valueOf(iii);
String periodo=String.valueOf(periodo);

/* INFORMACIÓN DE OPERADOR DE SALUD 1 */
ArrayList<String> aa1=new ArrayList<String>(a1.size());
for(Integer m:a1){
    aa1.add(String.valueOf(m));
}
ArrayList<String> bb1=new ArrayList<String>(b1.size());
for(Double n:b1){
    bb1.add(String.valueOf(n));
}

ArrayList<String> cc1=new ArrayList<String>(aa1.size()+bb1.size());
cc1.addAll(aa1);
cc1.addAll(bb1);

String [] dd1 = new String [cc1.size()];
dd1=cc1.toArray(dd1);

String ee1=Arrays.toString(dd1)+iii+"_"+periodo;

ACLMessage OP1= new ACLMessage(ACLMessage.INFORM);
OP1.setContent(ee1);
OP1.addReceiver(new AID("AgenteOperador1",AID.ISLOCALNAME));
send (OP1);

/* INFORMACIÓN DE OPERADOR DE SALUD 2 */

ArrayList<String> aa2=new ArrayList<String>(a2.size());
for(Integer m:a2){
    aa2.add(String.valueOf(m));
}
ArrayList<String> bb2=new ArrayList<String>(b2.size());
for(Double n:b2){
    bb2.add(String.valueOf(n));
}

ArrayList<String> cc2=new ArrayList<String>(aa2.size()+bb2.size());
cc2.addAll(aa2);
cc2.addAll(bb2);

String [] dd2 = new String [cc2.size()];
dd2=cc2.toArray(dd2);

String ee2=Arrays.toString(dd2)+iii+"_"+periodo;

ACLMessage OP2= new ACLMessage(ACLMessage.INFORM);
OP2.setContent(ee2);
OP2.addReceiver(new AID("AgenteOperador2",AID.ISLOCALNAME));
send (OP2);

/* INFORMACIÓN DE OPERADOR DE SALUD 3 */

```

MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA

```
ArrayList<String> aa3=new ArrayList<String>(a3.size());
    for(Integer m:a3){
        aa3.add(String.valueOf(m));
    }
ArrayList<String> bb3=new ArrayList<String>(b3.size());
for(Double n:b3){
    bb3.add(String.valueOf(n));
}

    ArrayList<String> cc3=new ArrayList<String>(aa3.size()+bb3.size());
cc3.addAll(aa3);
cc3.addAll(bb3);

    String [] dd3 = new String [cc3.size()];
dd3=cc3.toArray(dd3);

    String ee3=Arrays.toString(dd3)+iiii+"_"+periodo;

    ACLMessage OP3= new ACLMessage(ACLMessage.INFORM);
OP3.setContent(ee3);
OP3.addReceiver(new AID("AgenteOperador3",AID.ISLOCALNAME));
send (OP3);

/* INFORMACIÓN DE OPERADOR DE SALUD 4 */

ArrayList<String> aa4=new ArrayList<String>(a4.size());
for(Integer m:a4){
    aa4.add(String.valueOf(m));
}
ArrayList<String> bb4=new ArrayList<String>(b4.size());
for(Double n:b4){
    bb4.add(String.valueOf(n));
}

    ArrayList<String> cc4=new ArrayList<String>(aa4.size()+bb4.size());
cc4.addAll(aa4);
cc4.addAll(bb4);

    String [] dd4 = new String [cc4.size()];
dd4=cc4.toArray(dd4);

    String ee4=Arrays.toString(dd4)+iiii+"_"+periodo;

    ACLMessage OP4= new ACLMessage(ACLMessage.INFORM);
OP4.setContent(ee4);
OP4.addReceiver(new AID("AgenteOperador4",AID.ISLOCALNAME));
send (OP4);

    } catch (XPRSexception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    }else
    {
        System.out.println("Esperando a recibir aviso");
    }
}
}

@Override
public boolean done() {
    // TODO Auto-generated method stub
    return false;
}
}}
```

Clase AgenteGenerador

```

package Agentes;

import java.util.*;
import java.util.Random;
import jsc.distributions.*;
import java.io.*;
import jade.core.Agent;
import jade.core.behaviours.*;
import jade.lang.acl.*;
import jade.core.AID;

public class AgenteGenerador extends Agent{
    static Random rand=new Random(99);
    int i;
    static long periodo=120000;
    protected void setup(){

        System.out.println("Hola, soy el agente "+getLocalName());
        addBehaviour(new Generador());
    }

    protected void takeDown(){
        System.out.println("Ejecucion finalizada "+getLocalName());
    }

    public class Generador extends SimpleBehaviour{

        @Override
        public void action() {
            double rand=Math.random();
            DiscreteUniform pp;
            pp=new DiscreteUniform(4,10);

            //int P=(int) pp.inverseCdf(rand);
            int P=8;
            Escribir("Datos"+i+".txt",P,i,periodo);

            String j;
            j=String.valueOf(i)+"_"+String.valueOf(periodo);
            System.out.println("-----");
            -----
            -----
            -----");

            System.out.println("*****
            *****");

            System.out.println("*****
            *****");
            System.out.println("-----");
            -----
            -----");

            System.out.println("Enviando información del sistema de cuidado a la salud
            domiciliaria para el periodo "+(i*periodo)+"-"+((i*periodo)+periodo));
            ACLMessage aviso= new ACLMessage(ACLMessage.INFORM);
            aviso.setContent(j);
            aviso.addReceiver(new AID("AgenteCoordinador",AID.ISLOCALNAME));
            send (aviso);
        }
        @Override
        public boolean done() {
            i++;
            if(i<9){

```

*MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL
PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA*

```

        block(periodo);
        return false;
    }else{
        System.out.println("Simulación Finalizada");

        return true;
    }
}

}

}

public void Escribir(String nombre,int P,int i,long periodo){
    int K=4;
    double gama=16.071;
    double sigma=-37.929;
    int epsilon=24;

    File f;
    FileWriter w;
    BufferedWriter bw;
    PrintWriter wr;
    try{
        f=new File(nombre);
        w=new FileWriter(f);
        bw=new BufferedWriter(w);
        wr=new PrintWriter(bw);
        int [] a=AgenteGenerador.Llegadasacu(P,i,periodo);
        int [] s=AgenteGenerador.servicios(P);
        int [] p=AgenteGenerador.prioridades(P);
        int [] h=AgenteGenerador.habilidadesP(P);
        double [] xp=AgenteGenerador.coordXpac(P);
        double [] yp=AgenteGenerador.coordYpac(P);
        int [] q=AgenteGenerador.habilidadesOP(K);
        double [] xop=AgenteGenerador.coordXOpe(P,i);
        double [] yop=AgenteGenerador.coordYOpe(P,i);
        int [] l=AgenteGenerador.inicios(P,i);

        wr.write("Datos\n");
        wr.write("K "+K+"\n");
        wr.write("P "+P+"\n");
        wr.write("gama "+gama+"\n");
        wr.write("sigma "+sigma+"\n");
        wr.write("epsilon "+epsilon+"\n");
        wr.write("paciente\n");
        wr.write("i !x !y !a !s !h !p\n");
        for(int e=0;e<K;e++){
            wr.write(e+" "+xop[e]+" "+yop[e]+" "+0+" "+0+" "+0+" "+0+"\n");
        }
        for(int j=0;j<P;j++){
            wr.write(j+K+" "+xp[j]+" "+yp[j]+" "+a[j]+" "+s[j]+" "+p[j]+" "+h[j]+"
"+ "\n");
        }
        wr.write("operadores\n");
        wr.write("K q l u+"\n");
        for(int e=0;e<K;e++){
            wr.write(e+" "+5+" "+l[e]+" "+800000+"\n");
        }

        wr.write("Estados de los operadores de salud después del ruteo\n");

        wr.close();
        bw.close();
    }catch (Exception e){
        System.out.println("Se ha producido un error :"+e);
    }
}

}

public static int[] Llegadasacu(int P,int i,long periodo){
    //Random rand=new Random(100);
    //Uniform ll;
    //ll= new Uniform(10000,12000);
    int ll=15000;

```

```

int [] arregloa=new int[P];
int [] sumarregloa=new int[P];
if(i==0){
    for (int j=0;j<arregloa.length;j++){
        //int a=(int) Math.rint(Il.inverseCdf(rand.nextDouble()));
        int a=ll;
        arregloa[j]=a;
    }
    for(int j=0;j<sumarregloa.length;j++){
        if(j==0){
            sumarregloa[j]=arregloa[j];
        }else
            sumarregloa[j]=arregloa[j]+sumarregloa[j-1];
    }
}
else{
    int m=(int) (i*periodo);
    //int a=(int) Math.rint(Il.inverseCdf(rand.nextDouble()));
    int a=ll;
    arregloa[0]=m+a;
    for (int j=1;j<arregloa.length;j++){
        arregloa[j]=a;
    }for(int j=0;j<sumarregloa.length;j++){
        if(j==0){
            sumarregloa[j]=arregloa[j];
        }else
            sumarregloa[j]=arregloa[j]+sumarregloa[j-1];
    }
}
return sumarregloa;
}
public static int[] servicios(int P){
    //Random rand=new Random(100);
    Uniform ss;
    ss=new Uniform(8000,10000);
    int [] arreglos=new int[P];
    for(int i=0;i<arreglos.length;i++){
        int s=(int) Math.rint(ss.inverseCdf(rand.nextDouble()));
        arreglos[i]=s;
    }
    return arreglos;
}
public static int[] prioridades(int P){
    //Random rand=new Random(80);
    DiscreteUniform pp;
    pp=new DiscreteUniform(2,4);
    int [] arreglop=new int[P];
    for(int i=0;i<arreglop.length;i++){
        int p=(int) pp.inverseCdf(rand.nextDouble());
        arreglop[i]=p;
    }
    return arreglop;
}
public static int[] habilidadesP(int P){
    //Random rand=new Random(100);
    DiscreteUniform hh;
    hh=new DiscreteUniform(1,3);
    int [] arregloh=new int[P];
    for(int i=0;i<arregloh.length;i++){
        int h=(int) hh.inverseCdf(rand.nextDouble());
        arregloh[i]=h;
    }
    return arregloh;
}
public static int[] habilidadesOP(int K){
    //Random rand=new Random(100);
    DiscreteUniform opp;
    opp=new DiscreteUniform(1,5);
    int [] arreglop=new int[K];
    for(int i=0;i<arreglop.length;i++){
        int op=(int) opp.inverseCdf(rand.nextDouble());

```

*MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL
PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA*

```

        arreglop[i]=op;
    }
    return arreglop;
}
public static double[] coordXpac(int P){
    //Random rand=new Random(100);
    Uniform xx;
    xx=new Uniform(-100,100);
    double [] arreglox=new double [P];
    for(int i=0;i<arreglox.length;i++){
        double x=(double) Math.rint((xx.inverseCdf(rand.nextDouble()))*1000)/1000;
        arreglox[i]=x;
    }
    return arreglox;
}
public static double[] coordYpac(int P){
    //Random rand=new Random(90);
    Uniform yy;
    yy=new Uniform(-100,100);
    double [] arregloy=new double [P];
    for(int i=0;i<arregloy.length;i++){
        double y=(double) Math.rint((yy.inverseCdf(rand.nextDouble()))*1000)/1000;
        arregloy[i]=y;
    }
    return arregloy;
}
public static double[] coordXOpe(int K,int i){
    //Random rand=new Random(110);
    Uniform xxx;
    xxx=new Uniform(-100,100);
    double [] arregloxx=new double [K];
    if(i==0){
        for(int j=0;j<arregloxx.length;j++){
            double xx=(double) Math.rint((xxx.inverseCdf(rand.nextDouble()))*1000)/1000;
            arregloxx[j]=xx;
        }
    }else{
        File f;
        f=new File("Datos"+(i-1)+".txt");
        Scanner scanner = null;
        try{
            scanner=new Scanner(f);
            List<String> list= new ArrayList<>();
            while(scanner.hasNextLine()){
                list.add(scanner.nextLine());
            }
            int rr1=Integer.valueOf(list.get(list.indexOf("OP1")+1));
            int rr2=Integer.valueOf(list.get(list.indexOf("OP2")+1));
            int rr3=Integer.valueOf(list.get(list.indexOf("OP3")+1));
            int rr4=Integer.valueOf(list.get(list.indexOf("OP4")+1));

            String [] Lista=new String[list.size()];
            Lista=list.toArray(Lista);

            String ss1=Lista[rr1+8];
            String ss2=Lista[rr2+8];
            String ss3=Lista[rr3+8];
            String ss4=Lista[rr4+8];

            String [] dividir1=ss1.split(" ");
            double [] tt1=new double [dividir1.length];
            for(int l=0;l<tt1.length;l++){
                tt1[l]=Double.parseDouble(dividir1[l]);
            }
            String [] dividir2=ss2.split(" ");
            double [] tt2=new double [dividir2.length];
            for(int l=0;l<tt1.length;l++){
                tt2[l]=Double.parseDouble(dividir2[l]);
            }
        }catch (Exception e){
            e.printStackTrace();
        }
    }
}

```

```

    }
    String [] dividir3=ss3.split(" ");
    double [] tt3=new double [dividir3.length];
    for(int l=0;l<tt3.length;++){
        tt3[l]=Double.parseDouble(dividir3[l]);
    }
    String [] dividir4=ss4.split(" ");
    double [] tt4=new double [dividir4.length];
    for(int l=0;l<tt4.length;++){
        tt4[l]=Double.parseDouble(dividir4[l]);
    }

    Double uu1=tt1[1];
    Double uu2=tt2[1];
    Double uu3=tt3[1];
    Double uu4=tt4[1];

    arregloxx[0]=uu1;
    arregloxx[1]=uu2;
    arregloxx[2]=uu3;
    arregloxx[3]=uu4;

} catch (Exception e){
    System.out.println("Se ha producido un error :"+e);
}
finally {
    if(scanner!=null)
        scanner.close();
}
}
return arregloxx;
}

public static double[] coordYOpe(int K,int i){
    //Random rand=new Random(110);
    Uniform yyyy;
    yyyy=new Uniform(-100,100);
    double [] arregloyy=new double [K];
    if(i==0){
        for(int j=0;j<arregloyy.length;j++){
            double yy=(double) Math rint((yyyy.inverseCdf(rand.nextDouble()))*1000)/1000;
            arregloyy[j]=yy;
        }
    }else{
        File f;
        f=new File("Datos"+(i-1)+".txt");
        Scanner scanner = null;
        try{
            scanner=new Scanner(f);
            List<String> list= new ArrayList<>();
            while(scanner.hasNextLine()){
                list.add(scanner.nextLine());
            }
            int rr1=Integer.valueOf(list.get(list.indexOf("OP1")+1));
            int rr2=Integer.valueOf(list.get(list.indexOf("OP2")+1));
            int rr3=Integer.valueOf(list.get(list.indexOf("OP3")+1));
            int rr4=Integer.valueOf(list.get(list.indexOf("OP4")+1));

            String [] Lista=new String[list.size()];
            Lista=list.toArray(Lista);

            String ss1=Lista[rr1+8];
            String ss2=Lista[rr2+8];
            String ss3=Lista[rr3+8];
            String ss4=Lista[rr4+8];

            String [] dividir1=ss1.split(" ");
            double [] tt1=new double [dividir1.length];
            for(int l=0;l<tt1.length;++){
                tt1[l]=Double.parseDouble(dividir1[l]);
            }
        }
    }
}

```

*MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL
PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA*

```
String [] dividir2=ss2.split(" ");
double [] tt2=new double [dividir2.length];
for(int l=0;l<tt1.length;++){
    tt2[l]=Double.parseDouble(dividir2[l]);
}
String [] dividir3=ss3.split(" ");
double [] tt3=new double [dividir3.length];
for(int l=0;l<tt3.length;++){
    tt3[l]=Double.parseDouble(dividir3[l]);
}
String [] dividir4=ss4.split(" ");
double [] tt4=new double [dividir4.length];
for(int l=0;l<tt4.length;++){
    tt4[l]=Double.parseDouble(dividir4[l]);
}

Double uu1=tt1[2];
Double uu2=tt2[2];
Double uu3=tt3[2];
Double uu4=tt4[2];

arregloyy[0]=uu1;
arregloyy[1]=uu2;
arregloyy[2]=uu3;
arregloyy[3]=uu4;

}catch (Exception e){
    System.out.println("Se ha producido un error :"+e);
}
finally {
    if(scanner!=null)
        scanner.close();
}
}
return arregloyy;
}
public static int [] inicios(int K,int i){

    int [] inici=new int [K];

    if(i==0){

        for(int c=0;c<inici.length;c++){
            inici[c]=(int) periodo;
        }
    }else{

        File f;
        f=new File("Datos"+(i-1)+".txt");
        Scanner scanner = null;
        try{
            scanner=new Scanner(f);
            List<String> lista= new ArrayList<>();
            while(scanner.hasNextLine()){
                lista.add(scanner.nextLine());
            }

            int tt1=Integer.valueOf(lista.get(lista.indexOf("OP1")+2));
            int tt2=Integer.valueOf(lista.get(lista.indexOf("OP2")+2));
            int tt3=Integer.valueOf(lista.get(lista.indexOf("OP3")+2));
            int tt4=Integer.valueOf(lista.get(lista.indexOf("OP4")+2));

            inici[0]=tt1;
            inici[1]=tt2;
            inici[2]=tt3;
            inici[3]=tt4;

        }catch (Exception e){
```

```

        System.out.println("Se ha producido un error :"+e);
    }
    finally {
        if(scanner!=null)
            scanner.close();
    }
}

return inici;
}
}

```

Clase AgenteOperador

```

package Agentes;
import jade.core.Agent;
import jade.core.behaviours.*;
import jade.lang.acl.*;
import java.util.*;
import java.io.*;
public class AgenteOperador1 extends Agent {
public void setup(){

        System.out.println("Hola, soy el agente"+getLocalName());

        addBehaviour(new AtenderPacientes());

    }
    protected void takeDown(){
        System.out.println("Ejecucion finalizada "+getLocalName());
    }

    public class AtenderPacientes extends SimpleBehaviour{
        public void action() {

            ACLMessage OP1 = receive();
            if(OP1!=null){

                String ff1=OP1.getContent();
                int sa = ff1.indexOf("[");
                int sb = ff1.indexOf("]");
                int sc = ff1.indexOf("_");
                String gg1=ff1.substring(sa+1,sb);
                String arch=ff1.substring(sb+1,sc);
                String per=ff1.substring(sc+1, ff1.length());

                int archivo=(int) Double.parseDouble(arch);
                int periodo=(int) Double.parseDouble(per);
                String[] hh1 = gg1.split(", ");
                double [] ii1 = new double[hh1.length];
                for(int i=0;i<ii1.length;i++){
                    ii1[i]=Double.parseDouble(hh1[i]);
                }
                double[] jj1=Arrays.copyOfRange(ii1,1, (ii1.length/2)-1);/*Rango desde 1 para quitar el
origen*/

                double[] kk1=Arrays.copyOfRange(ii1,(ii1.length/2)+1, ii1.length-1); /*Rango hasta -1
para quitar el destino*/

                int ll1;
                int mm1 = 0;
                int kk=archivo*periodo;

                if(archivo==0){
                    if(jj1.length>0){
                        ll1=(int) jj1[jj1.length-1];

                        File f;

```

MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA

```
f=new File("Datos"+archivo+".txt");
Scanner scanner = null;
try{
    scanner=new Scanner(f);
    List<String> list= new ArrayList<>();
    while(scanner.hasNextLine()){
        list.add(scanner.nextLine());
    }
    scanner.close();
    String [] Lista=new String[list.size()];
    Lista=list.toArray(Lista);

    String zz1=Lista[ll1+8];

    String [] dividir1=zz1.split(" ");
    double [] aaa1=new double [dividir1.length];
    for(int l=0;l<aaa1.length;l++){
        aaa1[l]=Double.parseDouble(dividir1[l]);
    }

    double serv=aaa1[4];
    mm1=(int) (kk1[kk1.length-1]+serv);

    System.out.println("Operador 1 va a los nodos
"+Arrays.toString(jj1)+" llega en el tiempo "+Arrays.toString(kk1)+" respectivamente y estará disponible en: "+mm1);
    if(mm1<periodo*2){
        mm1=periodo*2;
    }
}catch (Exception e){
    System.out.println("Se ha producido un error :"+e);
}

}else{

    ll1=0;
    mm1=periodo*2;
    System.out.println("Operador 1 no cambia ubicación");
}

}else{
    if(jj1.length>0){

        ll1=(int) jj1[jj1.length-1];

        File f;
        f=new File("Datos"+archivo+".txt");
        Scanner scanner = null;
        try{

            scanner=new Scanner(f);
            List<String> list= new ArrayList<>();
            while(scanner.hasNextLine()){
                list.add(scanner.nextLine());
            }
            scanner.close();
            String [] Lista=new String[list.size()];
            Lista=list.toArray(Lista);

            String zz1=Lista[ll1+8];

            String [] dividir1=zz1.split(" ");
            double [] aaa1=new double [dividir1.length];
            for(int l=0;l<aaa1.length;l++){
                aaa1[l]=Double.parseDouble(dividir1[l]);
            }

            double serv=aaa1[4];
            mm1=(int) (kk1[kk1.length-1]+serv);
```


MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA

```
import java.io.FileReader;
import java.util.StringTokenizer;

public class LeerDatos {

    /***DATA***/

    /***SETS***/
    int K;    /* Number of Operators*/
    int N;    /* Number of Nodes*/
    int P;    /* Number of Patients*/
    int KI;  /* Number of Depots*/

    /***DATA***/
    double [] Nodes;    /* Nodes */
    double[] Depots;    /* Nodes */
    double[] cx;        /* x */
    double[] cy;        /* y */
    double[] a;         /* lower bound of time windows*/
    double[] s;         /* service time*/
    double[] h;         /* skills of operators*/
    double[] p;         /* triage */
    double [] q;        /* capacity */
    double [] l;        /* lower bound of depot */
    double [] u;        /* upper bound of depot */
    double gama;        /* parameter of triage */
    double sigma;       /* parameter of triage */
    double epsilon;     /* parameter of triage */

    /***OTHERS***/
    BufferedReader input;
    String name,fc;

    public LeerDatos(String Instance){

        try{
            input = new BufferedReader (new FileReader (Instance));
            StringTokenizer line = new StringTokenizer (input.readLine ());
            name=line.nextToken();
            line = new StringTokenizer (input.readLine ());
            fc=line.nextToken();
            K=Integer.parseInt(line.nextToken());
            line = new StringTokenizer (input.readLine ());
            fc=line.nextToken();
            P=Integer.parseInt(line.nextToken());
            line = new StringTokenizer (input.readLine ());
            fc=line.nextToken();
            gama=Double.parseDouble(line.nextToken());
            line = new StringTokenizer (input.readLine ());
            fc=line.nextToken();
            sigma=Double.parseDouble(line.nextToken());
            line = new StringTokenizer (input.readLine ());
            fc=line.nextToken();
            epsilon=Double.parseDouble(line.nextToken());
            line = new StringTokenizer (input.readLine ());
            fc=line.nextToken();
            line = new StringTokenizer (input.readLine ());
            fc=line.nextToken();
            Nodes = new double[P+K];
            cx=new double[P+K];
            cy=new double[P+K];
            a=new double[P+K];
            s=new double[P+K];
            h=new double[P+K];
            p=new double[P+K];
            for (int i=0;j<=P+K-1;i++){
                line = new StringTokenizer (input.readLine());
                Nodes[i]=Integer.parseInt(line.nextToken());
                cx[i]=Double.parseDouble(line.nextToken());
```

```

        cy[i]=Double.parseDouble(line.nextToken());
        a[i]=Double.parseDouble(line.nextToken());
        s[i]=Double.parseDouble(line.nextToken());
        h[i]=Double.parseDouble(line.nextToken());
        p[i]=Double.parseDouble(line.nextToken());
    }

    line = new StringTokenizer (input.readLine ());
    line = new StringTokenizer (input.readLine ());
    Depots=new double[K];
    q=new double[K];
    l=new double[K];
    u=new double[K];
    for (int i=0;j<=K-1;i++){
        line = new StringTokenizer (input.readLine());
        Depots[i]=Integer.parseInt(line.nextToken());
        q[i]=Double.parseDouble(line.nextToken());
        l[i]=Double.parseDouble(line.nextToken());
        u[i]=Double.parseDouble(line.nextToken());
    };
    input.close ();
} catch (Exception e){
    System.out.println("Se ha producido un error :"+e);
}
}
}
}

```

Clase Modelomat

```

package Agentes;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;
import com.dashoptimization.XPRB;
import com.dashoptimization.XPRBctr;
import com.dashoptimization.XPRBexpr;
import com.dashoptimization.XPRBprob;
import com.dashoptimization.XPRBvar;
import com.dashoptimization.XPRSexception;
public class Modelomatemat {
    /***VARIABLES*****/
    static XPRBvar[][][] X; /* Visit Variables */
    static XPRBvar[][] Y; /* Start time service */
    static XPRBvar[] Z; /* service promise time */
    /***CONSTRAINTS*****/
    static XPRBctr[] visit; /*Constraint to each patient is served once*/
    static XPRBctr[] outdepot; /*Constraint to each operator leaves on depot*/
    static XPRBctr[] outdepot1; /*Constraint to each operator does not leave other depots*/
    static XPRBctr[] indepot; /*Constraint to each operator return on depot*/
    static XPRBctr[] indepot1; /*Constraint to each operator does not return other depot*/
    static XPRBctr[][] balance; /*Constraint ensure that a operator arrives in each patient leaves
this*/
    static XPRBctr[][][] secuencing; /*Constraint to visit in each patient*/
    static XPRBctr[][] lbTimeWindow; /*Constraint to lower bound of time window*/
    static XPRBctr[] triage; /*Constraint to triage*/
    static XPRBctr[] service; /*Constraint to service promise*/
    static XPRBctr[][] skills; /*Constraint to skills for operator*/
    static XPRBctr[][] Starlaborday; /*Constraint to start labor day*/
    static XPRBctr[][] Endlaborday; /*Constraint to end labor day*/
    /***CONSTRAINTS*****/
    static XPRB bcl; /*new problem*/
    static XPRBprob pro; /*new problem*/
    static XPRBctr cobj; /* Objective function */
    ArrayList<ArrayList<Integer>> path;
    ArrayList<ArrayList<Double>> pathst;
    ArrayList<Double> yy;
    ArrayList<Double> zz;
    private double[] u;
    private double[] l;
}

```

*MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL
PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA*

```

private double[] q;
private double[] p;
private double[] h;
private double[] s;
private double[] a;
private double[][] t;
public Modelomatemat(double alpha,double beta,int K,int N,int P,double[] Nodes,double[] Depots,double[] cx,double[]
cy,double[] a,double[] s,double[] h,double[] p,double [] q,double [] l,double [] u,double gama,double sigma,double epsilon) throws
XPRSexception, IOException{
    double computetime; //Time
    computetime=XPRB.getTime());

    //AGREGAR FORMULA Y FACTOR DE VELOCIDAD
    double[][] t = new double [N][N];
    for (int i=0;i<K;i++){
        for (int j=K;j<K+P;j++){
            t[i][j]=Math.abs(cx[i]-cx[j])+Math.abs(cy[i]-cy[j]);

            t[j][K+P+i]=Math.abs(cx[i]-cx[j])+Math.abs(cy[i]-cy[j]);
        }
    }
    for (int i=K;i<K+P;i++){
        for (int j=K;j<K+P;j++){
            if (i!=j) {
                t[i][j]=Math.abs(cx[i]-cx[j])+Math.abs(cy[i]-cy[j]);
            }
        }
    }

    System.out.println("Read Data Time "+(XPRB.getTime()-computetime)/1000+" seg");
    System.out.println("");
    System.out.println("Start of Routing Model");

    ArrayList<ArrayList<Double>> pathst=new ArrayList<ArrayList<Double>>();
    ArrayList<Double> zz=new ArrayList<Double>();
    ArrayList<Double> yy=new ArrayList<Double>();

    path=(modelroutingHHC(alpha,beta,pathst,zz,yy,t,a,s,h,p,q,l,u,N,K,P,gama,sigma,epsilon));
    System.out.println("Computational Time "+(XPRB.getTime()-computetime)/1000+" seg");
    /*** End-Routing model***/

}

/****
 * @ Routing model
 *
 * ***/
@SuppressWarnings("deprecation")
public ArrayList<ArrayList<Integer>> modelroutingHHC(double alpha, double beta,
    ArrayList<ArrayList<Double>> pathst,ArrayList<Double> zz,ArrayList<Double> yy,
    double [][]t,double [] a,double [] s,
    double[] h,double [] p, double [] q, double [] l, double[] u,int N,int K,int P,double gama,double
sigma, double epsilon) throws XPRSexception, IOException{

    this.pathst=pathst;
    this.zz=zz;
    this.yy=yy;
    this.t=t;
    this.a=a;
    this.s=s;
    this.h=h;
    this.p=p;
    this.q=q;
    this.l=l;
    this.u=u;

    /*** problem definition***/

```

```

bcl= new XPRB();
pro=bcl.newProb("Routing");

/****VARIABLES****/

X = new XPRBvar[N][N][K];
Y = new XPRBvar[N][K];
Z= new XPRBvar[P+K];

for(int i=0;i<N;i++){
    for (int j=0;j<N;j++){
        for (int k=0;k<K;k++){
            if (i!=j){
                X[i][j][k]=pro.newVar("X_"+(i)+(j)+(k), XPRB.BV, 0,1);
            }
        }
    }
}

for(int i=0;i<N;i++){
    for (int k=0;k<K;k++){
        Y[i][k]=pro.newVar("Y_"+(i)+(k), XPRB.PL,0,XPRB.INFINITY);
    }
}

for(int i=K;i<K+P;i++){
    Z[i]=pro.newVar("Z_"+(i),XPRB.PL,0,XPRB.INFINITY);
}

/****OBJECTIVE****/
XPRBexpr le;
le = new XPRBexpr();
for(int i=0;i<N;i++){
    for(int j=0;j<N;j++){
        for(int k=0;k<K;k++){
            if (i!=j)le.add(X[i][j][k].mul(alpha*t[i][j]));
        }
    }
}

for(int i=K;i<K+P;i++){
    le.add(Z[i].mul(beta/(500*P)));
}

cobj = pro.newCtr("OBJ", le);
pro.setObj(cobj);      /* Minimize total Cost */

/****CONSTRAINTS****/
visit=new XPRBctr[P+K];
for(int j=K;j<P+K;j++)    /* Satisfy the demand per patient */
{
    le = new XPRBexpr();
    for(int i=0;i<P+K;i++){
        for(int k=0;k<K;k++){
            if(i!=j)le.add(X[i][j][k]);
        }
    }
    visit[j] = pro.newCtr("Visit_"+(j), le.eq(1));
}

balance=new XPRBctr[P+K][K];
for(int k=0;k<K;k++){    /* Flow balance per patient */
    for (int e=K;e<P+K;e++){
        le = new XPRBexpr();
        for(int i=0;i<K+P;i++){
            if (i!=e)le .add(X[i][e][k]);
        }
        for(int j=K;j<N;j++){
            if(j!=e)le .add(X[e][j][k].mul(-1));
        }
        balance[e][k] = pro.newCtr("Balance_"+"(e)+(k), le.eq(0));
    }
}

outdepot=new XPRBctr[K];
for (int k=0;k<K;k++){
    le = new XPRBexpr();
    for (int j=K;j<N;j++){
        le.add(X[k][j][k].mul(1));
    }
}

```

MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA

```

    }
    outdepot[k]=pro.newCtr("Outdepot_"+(k),le.eql(1));
}
outdepot1=new XPRBctr[K];
for (int k=0;k<K;k++){
    le = new XPRBexpr();
    for (int j=K;j<N;j++){
        for (int i=0;i<K;i++){
            if (i!=k){le.add(X[i][j][k].mul(1));}
        }
    }
    outdepot1[k]=pro.newCtr("Outdepot1_"+(k),le.eql(0));
}
indepot=new XPRBctr[K];
for (int k=0;k<K;k++){
    le = new XPRBexpr();
    for (int i=0;i<P+K;i++){
        le.add(X[i][k+P+K][k].mul(1));
    }
    indepot[k]=pro.newCtr("Indepot_"+(k),le.eql(1));
}
indepot1=new XPRBctr[K];
for (int k=0;k<K;k++){
    le = new XPRBexpr();
    for (int i=0;i<P+K;i++){
        for (int j=P+K;j<N;j++){
            if (j!=k+P+K){le.add(X[i][j][k].mul(1));}
        }
    }
    indepot1[k]=pro.newCtr("Indepot1_"+(k),le.eql(0));
}
secuencing = new XPRBctr[K][N][N];
double MM=100000;
for(int k=0;k<K;k++){
    for(int i=K;i<K+P;i++){
        for(int j=K;j<K+P;j++){
            if (i!=j){
                le = new XPRBexpr();
                le.add(Y[i][k]);
                le.add(Y[j][k].mul(-1));
                le.add(X[i][j][k].mul(MM));
            }
        }
        secuencing[k][i][j]=pro.newCtr("secuencingPat_"+(k)+(i)+(j),le.lEq(MM-t[i][j]-s[i]));
    }
}
for(int i=0;i<K;i++){
    for(int j=K;j<N;j++){
        if (i!=j){
            le = new XPRBexpr();
            le.add(Y[i][k]);
            le.add(Y[j][k].mul(-1));
            le.add(X[i][j][k].mul(MM));
        }
    }
    secuencing[k][i][j]=pro.newCtr("secuencingStDep_"+(k)+(i)+(j),le.lEq(MM-t[i][j]));
}
for(int i=0;i<K+P;i++){
    for(int j=K+P;j<N;j++){
        if (i!=j){
            le = new XPRBexpr();
            le.add(Y[i][k]);
            le.add(Y[j][k].mul(-1));
            le.add(X[i][j][k].mul(MM));
        }
    }
    secuencing[k][i][j]=pro.newCtr("secuencingEnDep_"+(k)+(i)+(j),le.lEq(MM-t[i][j]));
}

```

```

    }
    }
}
lbTimeWindow=new XPRBctr[K][K+P];
for(int k=0;k<K;k++){
    for(int i=K;i<K+P;i++){
        le = new XPRBexpr();
        le.add(Y[i][k]);
        for (int j=0;j<N;j++){
            if (i!=j)le.add(X[i][j][k].mul(-1*a[i]));
        }
        lbTimeWindow[k][i]=pro.newCtr("lbTimeWindow_"+(k)+(i),le.gEq(0));
    }
}
trriage=new XPRBctr[K+P];
for(int i=K;i<K+P;i++){
    le = new XPRBexpr();
    le.add(Z[i]);
    triage[i]=pro.newCtr("trriage_"+(i),le.gEq(gama*Math.pow(p[i],2)+sigma*p[i]+epsilon+a[i]));
}
service=new XPRBctr[K][K+P];
for (int k=0;k<K;k++){
    for(int i=K;i<K+P;i++){
        le = new XPRBexpr();
        le.add(Y[i][k]);
        le.add(Z[i].mul(-1));
        service[k][i]=pro.newCtr("Service_"+(k)+(i),le.lEq(0));
    }
}
skills=new XPRBctr[K][N][N];
for (int k=0;k<K;k++){
    for(int i=0;i<K+P;i++){
        for (int j=K;j<K+P;j++){
            le = new XPRBexpr();
            if (i!=j)le.add(X[i][j][k].mul(MM));
            skills[k][i][j]=pro.newCtr("Skills_"+(k)+(i)+(j),le.lEq(q[k]-h[j]+MM));
        }
    }
}
Starlabyrinth=new XPRBctr[K][K];
for (int k=0;k<K;k++){
    for(int i=0;i<K;i++){
        le = new XPRBexpr();
        le.add(Y[i][k]);
        Starlabyrinth[k][i]=pro.newCtr("Starlabyrinth"+(k)+(i),le.gEq(l[i]));
    }
}
Endlabyrinth=new XPRBctr[K][N];
for (int k=0;k<K;k++){
    for(int j=K+P;j<N;j++){
        le = new XPRBexpr();
        le.add(Y[j][k]);
        Endlabyrinth[k][j]=pro.newCtr("Endlabyrinth"+(k)+(j),le.lEq(u[j]-K-P));
    }
}

/****SOLVING****/
pro.setSense(XPRB.MINIM);      /* Choose the sense of the optimization */
pro.solve("g");                /* Solve the MIP-problem */
System.out.println("Objective: " + pro.getObjVal()); /* Get objective value */
pro.exportProb("Folio.txt");

/****OUTPUT OF ROUTING MODEL****/

ArrayList<ArrayList<Integer>> path=new ArrayList<ArrayList<Integer>>();

for(int j=0;j<K+P;j++)zz.add(0.0);
for(int j=0;j<N;j++)yy.add(0.0);
for (int k=0;k<K;k++){

```

MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA

```

int i=k;
ArrayList<Integer> subpath = new ArrayList<Integer>();
ArrayList<Double> subpathst=new ArrayList<Double>();
subpath.add(k);
subpathst.add(Y[k][k].getSol());
path.add(subpath);
pathst.add(subpathst);
yy.set(k, Y[k][k].getSol());
while(i!=k+P+K){
    for(int j=K;j<N;j++){
        if(i!=j && X[i][j][k].getSol()>=0.0001){
            if (j<P+K){
                zz.set(j,Z[j].getSol());
            }
            subpath.add(j);
            subpathst.add(Y[j][k].getSol());
            yy.set(j, Y[j][k].getSol());
            i=j;
            j=N+1;
        }
    }
}
double costruteo=0;
for(int i=0;i<N;i++){
    for(int j=0;j<N;j++){
        for(int k=0;k<K;k++){
            if (i=j) costruteo=costruteo+X[i][j][k].getSol()*alpha*t[i][j];
        }
    }
}
System.out.println("CostoRuteo: "+costruteo);
double costpromesa=0;
for(int i=K;i<K+P;i++){
    costpromesa=costpromesa+Z[i].getSol()*beta/(500*P);
}
System.out.println("CostoPromesa: "+costpromesa);

return path;
}
}

```

Anexo 4: información histórica de solicitudes por horas

Tabla 38 Información histórica de las solicitudes por horas

D Í A	0	1	2	3	4	5	6	7	8	9	0	1	1	1	1	1	1	1	1	1	1	2	2	2	2
1	0	0	0	0	0	0	0	1	1	1	1	0	3	0	0	2	0	0	3	2	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	8	2	2	4	5	1	1	3	3	1	2	2	1	0	0	0
3	0	0	0	1	0	5	6	2	4	2	2	2	4	5	1	1	3	3	3	3	2	3	1	1	1
4	0	1	0	0	1	1	7	2	4	1	1	0	2	2	3	3	4	3	4	3	2	2	0	0	0
5	0	0	0	0	1	2	5	1	4	2	3	1	0	2	4	3	1	4	3	3	3	2	0	0	0
6	0	1	0	1	0	3	4	2	5	3	3	3	0	2	1	1	1	2	3	1	5	1	0	0	0
7	0	0	0	1	0	0	2	1	0	2	0	0	0	0	0	2	2	2	2	2	0	1	2	0	0
8	0	0	1	0	0	0	1	2	2	2	4	3	2	0	0	3	3	2	3	2	2	0	0	0	0
9	0	0	0	0	0	1	2	3	0	1	2	1	2	1	2	0	2	3	1	4	2	2	0	0	0
10	0	0	0	0	0	1	2	6	5	1	4	2	1	1	2	2	2	4	4	3	3	0	1	1	1
11	0	0	0	0	1	0	3	3	4	4	1	1	3	2	2	4	4	3	4	5	3	0	0	0	0
12	0	0	2	0	0	1	4	4	5	0	3	1	1	0	4	1	4	3	0	4	2	2	1	1	1

ANEXOS

13	0	0	1	0	1	0	3	2	5	2	2	1	4	1	2	0	2	3	5	0	0	1	0	1
14	2	0	0	0	2	0	0	4	0	3	2	0	1	1	1	3	0	1	1	2	0	0	1	0
15	0	0	1	1	0	1	0	2	2	1	4	2	1	1	2	3	1	4	0	0	2	1	1	0
16	1	0	1	0	0	2	2	1	2	1	1	4	2	3	2	0	3	1	4	3	2	3	0	1
17	0	0	0	0	0	3	4	1	2	2	0	7	2	0	5	4	6	4	3	2	0	0	3	0
18	0	2	0	0	0	1	3	4	2	4	2	2	3	3	4	1	4	1	3	1	5	0	1	0
19	0	0	0	0	0	0	2	1	0	3	3	0	1	2	2	1	1	4	3	0	1	1	1	0
20	0	0	1	0	1	1	2	5	2	1	0	1	0	3	3	0	2	2	1	4	1	1	1	0
21	1	0	0	0	0	1	0	1	3	4	1	0	0	0	0	1	1	2	0	1	1	0	0	0
22	0	0	0	1	0	0	0	3	1	0	2	1	1	0	0	1	1	4	1	2	1	0	1	0
23	0	0	0	0	0	1	1	3	4	3	1	0	1	2	2	4	1	0	4	0	1	1	1	0
24	0	0	0	0	0	0	0	3	4	0	1	9	3	2	2	1	0	1	0	1	2	1	0	0
25	0	0	0	0	0	3	3	1	5	5	0	1	5	2	1	5	1	2	2	3	2	0	0	0
26	1	0	0	0	0	3	0	3	3	5	1	4	2	4	3	0	0	1	3	1	3	1	1	0
27	0	0	0	0	0	0	1	1	3	3	4	0	0	2	1	1	0	3	1	3	2	0	1	0
28	0	0	0	0	0	0	1	1	0	5	0	1	0	1	0	0	0	2	0	3	0	0	1	0
29	0	0	0	0	0	0	1	0	6	3	0	0	0	0	0	3	0	0	1	0	1	1	2	0
30	0	0	0	0	0	0	1	2	2	2	0	1	0	1	3	1	2	0	0	4	0	2	0	0
31	0	0	1	0	0	1	4	4	3	1	4	1	3	0	2	4	2	3	3	3	1	0	1	0
32	0	0	0	0	0	2	3	2	2	8	0	0	2	2	3	3	2	3	5	2	0	2	2	0
33	0	0	0	0	0	0	4	3	6	1	2	0	1	0	1	2	3	0	1	4	0	3	1	0
34	0	0	0	1	2	0	1	4	1	3	2	2	2	0	0	1	3	0	0	0	0	2	0	0
35	0	1	0	0	0	1	1	0	5	1	1	4	1	0	0	1	1	1	0	2	3	2	0	1
36	0	0	1	0	0	1	1	2	3	1	3	2	1	3	0	1	0	0	2	2	1	0	0	0
37	0	1	0	0	0	0	1	6	3	4	2	1	2	1	1	1	4	2	6	2	2	2	0	1
38	0	0	0	0	1	2	2	2	2	3	3	0	0	0	2	1	5	3	3	1	2	3	1	1
39	1	0	1	2	0	2	2	0	2	4	0	0	0	1	4	2	2	5	3	5	3	3	0	0
40	0	0	0	0	3	0	3	0	2	3	4	1	2	3	0	6	4	1	1	4	2	2	0	1
41	1	0	0	0	1	1	4	5	3	0	1	1	0	1	1	1	1	5	3	0	1	1	0	0
42	0	0	0	0	0	0	1	2	0	1	0	1	2	2	1	0	2	2	1	1	0	2	0	0
43	0	0	0	1	1	1	0	1	2	2	3	0	1	1	0	3	1	1	0	1	0	0	0	0
44	0	0	0	0	0	2	5	6	4	6	2	1	2	1	2	3	1	1	5	4	3	0	0	0
45	0	0	0	0	0	0	1	2	3	0	3	3	0	5	2	0	4	3	5	3	3	1	2	0
46	0	0	0	0	0	1	2	4	2	4	2	1	2	1	0	1	3	2	3	3	1	1	0	0
47	1	0	0	0	0	1	4	1	2	4	1	1	1	2	0	0	3	1	3	4	2	0	0	0
48	1	0	0	0	1	1	2	1	2	3	4	2	1	1	0	1	4	0	1	0	2	0	2	2
49	0	0	0	0	0	0	3	0	1	3	2	1	0	1	1	0	2	1	1	2	0	1	0	1
50	0	1	0	0	0	0	3	2	1	1	1	0	0	1	0	1	1	1	3	2	4	1	0	0
51	0	0	0	0	0	1	4	0	0	0	0	0	$\frac{1}{9}$	5	1	3	2	8	2	1	1	0	0	0
52	0	0	0	0	1	2	3	4	4	1	2	4	3	1	3	3	2	5	3	1	1	0	0	0
53	0	1	0	0	1	0	0	3	2	2	2	1	2	2	0	1	3	3	3	2	3	1	2	0
54	1	0	0	0	1	0	1	3	2	1	2	1	3	0	2	4	5	4	5	2	1	0	0	0
55	1	0	0	0	0	1	0	6	1	0	0	2	4	1	0	2	3	3	2	0	1	3	2	0
56	0	1	0	0	1	0	2	0	4	1	1	3	0	2	0	1	2	1	1	2	1	1	0	0

*MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL
PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA*

57	1	0	2	0	0	1	0	3	1	1	0	0	2	2	1	1	2	0	4	1	0	0	0	1
58	0	0	1	0	0	1	3	5	3	3	0	0	0	0	1	1	1	0	6	3	2	3	0	1
59	0	0	0	0	0	0	7	0	0	0	0	0	0	1	0	1	2	2	5	5	1	2	2	0
60	0	0	0	1	1	3	1	0	0	0	0	0	0	0	1	0	0	2	3	1	2	1	0	0
61	0	0	0	0	0	1	2	1	0	0	1	0	0	0	1	3	0	0	5	3	3	0	0	1
62	1	0	0	0	0	3	5	3	0	0	0	0	0	0	2	1	2	2	2	3	2	0	1	0
63	0	0	0	0	0	1	1	2	2	3	4	1	3	0	0	2	1	3	1	2	1	2	1	0
64	0	0	0	0	2	0	2	2	3	3	3	1	1	2	1	3	2	2	2	0	0	0	1	0
65	0	1	0	0	1	0	$\frac{1}{1}$	5	3	2	2	4	2	0	5	7	5	2	2	4	0	0	2	0
66	0	0	0	1	0	0	3	1	3	1	4	3	3	0	0	0	2	1	4	0	1	0	1	0
67	0	0	0	0	0	1	2	2	3	2	4	4	3	0	0	0	0	3	0	4	1	0	1	0
68	0	0	0	0	0	3	1	0	4	1	3	0	2	0	1	0	1	2	1	0	1	0	1	0
69	0	0	0	0	1	1	6	2	1	1	2	1	4	1	4	1	2	2	2	1	4	2	0	0
70	0	0	0	0	0	0	0	2	5	1	0	1	1	0	1	0	0	0	2	1	0	1	0	0
71	0	0	0	0	0	0	2	1	3	1	3	2	1	1	1	1	0	4	3	1	1	2	0	0
72	0	2	1	2	1	3	0	0	7	4	2	1	3	5	1	2	3	5	2	3	0	0	0	0
73	0	1	0	0	0	0	1	5	3	3	2	1	0	3	1	3	2	3	3	4	1	2	3	0
74	1	0	0	0	0	0	3	7	4	2	2	3	3	0	1	0	5	1	3	3	5	1	0	1
75	0	1	0	0	1	0	3	4	6	1	2	1	1	4	2	3	3	2	3	1	0	0	2	0
76	1	0	0	0	2	0	0	6	6	1	4	0	4	0	1	0	2	1	1	1	2	0	0	0
77	0	0	0	0	0	0	1	2	2	3	3	0	0	2	2	3	0	0	1	1	2	0	0	0
78	0	0	0	0	0	0	0	2	3	0	0	1	1	1	2	0	1	0	0	1	0	1	0	0
79	0	0	0	0	0	0	0	0	3	1	0	2	2	3	0	3	3	2	2	2	1	1	0	1
80	0	0	0	0	1	0	5	4	1	1	3	2	4	3	0	3	1	3	3	3	3	2	1	0
81	0	0	0	0	2	2	0	3	5	3	3	2	3	1	1	0	3	2	4	3	1	1	1	1
82	0	1	0	0	2	0	2	1	4	6	2	2	1	1	1	3	2	3	3	6	2	0	0	1
83	0	0	0	0	0	0	1	0	1	5	5	1	2	1	1	1	1	2	3	2	1	2	1	1
84	1	0	2	0	0	0	0	0	2	1	1	3	0	1	0	1	3	2	1	1	1	0	0	0
85	1	1	0	0	0	1	0	4	5	4	3	0	2	0	3	2	2	4	1	2	1	1	0	1
86	0	1	0	0	0	3	1	4	3	4	2	1	3	3	4	3	3	3	3	4	2	0	0	0
87	0	0	0	1	0	1	5	6	3	2	3	2	2	3	1	1	6	3	1	0	2	1	0	0
88	1	1	0	0	1	0	1	5	6	3	4	3	1	1	1	2	3	2	2	0	0	0	0	1
89	0	0	1	0	1	3	1	5	2	0	3	1	4	0	0	0	0	0	1	3	3	0	1	0
90	0	0	1	0	1	1	7	6	2	2	1	2	0	0	0	0	0	1	0	3	2	2	0	0
91	1	1	0	1	0	1	0	3	3	2	0	1	2	3	0	1	3	1	2	3	0	0	1	0
92	0	0	0	0	1	1	0	0	0	0	4	2	0	1	1	1	2	1	3	4	3	2	0	0

Anexo 5: Análisis de entrada –Bondad de ajuste de las solicitudes por horas

0-1		1-2	
Binomial		Geometric	
n =	2.	p =	0.828829
p =	0.103261	Chi Squared	
Chi Squared		GOODNESS OF FIT: too few intervals for a good chi-squared test	
Kolmogorov-Smirnov		Kolmogorov-Smirnov	
data points	92	data points	92
ks stat	2.07e-004	ks stat	1.36e-002
alpha	5.e-002	alpha	5.e-002
ks stat(92,5.e-002)	0.14	ks stat(92,5.e-002)	0.14
p-value	1.	p-value	1.
result	DO NOT REJECT	result	DO NOT REJECT
2-3		3-4	
Geometric		Geometric	
p =	0.836364	p =	0.859813
Chi Squared		Chi Squared	
GOODNESS OF FIT: too few intervals for a good chi-squared test		GOODNESS OF FIT: too few intervals for a good chi-squared test	
Kolmogorov-Smirnov		Kolmogorov-Smirnov	
data points	92	data points	92
ks stat	5.83e-003	ks stat	2.76e-003
alpha	5.e-002	alpha	5.e-002
ks stat(92,5.e-002)	0.14	ks stat(92,5.e-002)	0.14
p-value	1.	p-value	1.
result	DO NOT REJECT	result	DO NOT REJECT
4-5		5-6	
Poisson		Poisson	
lamda =	0.413043	lamda =	0.880435
Chi Squared		Chi Squared	
GOODNESS OF FIT: too few intervals for a good chi-squared test		total classes	6
Kolmogorov-Smirnov		interval type	equal probable
data points	92	net bins	3
ks stat	1.23e-002	chi**2	1.08
alpha	5.e-002	degrees of freedom	2
ks stat(92,5.e-002)	0.14	alpha	5.e-002
p-value	1.	chi**2(2,5.e-002)	5.99
result	DO NOT REJECT	p-value	0.583
6-7		7-8	
Geometric		Kolmogorov-Smirnov	
p =	0.321678	data points	92
Chi Squared		ks stat	5.99e-002
total classes	7	alpha	5.e-002
interval type	equal probable	ks stat(92,5.e-002)	0.14
net bins	5	p-value	0.876
chi**2	4.87	result	DO NOT REJECT
degrees of freedom	4	7-8	
alpha	5.e-002	Poisson	
chi**2(4,5.e-002)	9.49	lamda =	2.43478
p-value	0.301	Chi Squared	
result	DO NOT REJECT	total classes	6
Kolmogorov-Smirnov		interval type	equal probable
data points	92	net bins	6
ks stat	8.25e-002	chi**2	14.6
alpha	5.e-002	degrees of freedom	5
ks stat(92,5.e-002)	0.14	alpha	5.e-002
p-value	0.53	chi**2(5,5.e-002)	11.1
result	DO NOT REJECT	p-value	1.23e-002
8-9		9-10	
Poisson		Kolmogorov-Smirnov	
lamda =	2.65217	data points	92
Chi Squared		ks stat	8.63e-002
total classes	6	alpha	5.e-002
interval type	equal probable	ks stat(92,5.e-002)	0.14
net bins	5	p-value	0.474
chi**2	0.96	result	DO NOT REJECT
degrees of freedom	4	9-10	
alpha	5.e-002	Poisson	
chi**2(4,5.e-002)	9.49	lamda =	2.16304
p-value	0.916	Chi Squared	
result	DO NOT REJECT	total classes	6
Kolmogorov-Smirnov		interval type	equal probable
data points	92	net bins	5
ks stat	7.08e-002	chi**2	5.95
alpha	5.e-002	degrees of freedom	4
ks stat(92,5.e-002)	0.14	alpha	5.e-002
p-value	0.719	chi**2(4,5.e-002)	9.49
result	DO NOT REJECT	p-value	0.203
9-10		9-10	
Poisson		Kolmogorov-Smirnov	
lamda =	2.16304	data points	92
Chi Squared		ks stat	8.2e-002
total classes	6	alpha	5.e-002
interval type	equal probable	ks stat(92,5.e-002)	0.14
net bins	5	p-value	0.539
chi**2	5.95	result	DO NOT REJECT
degrees of freedom	4	9-10	
alpha	5.e-002	Poisson	
chi**2(4,5.e-002)	9.49	lamda =	2.16304
p-value	0.203	Chi Squared	
result	DO NOT REJECT	total classes	6
Kolmogorov-Smirnov		interval type	equal probable
data points	92	net bins	5
ks stat	8.2e-002	chi**2	5.95
alpha	5.e-002	degrees of freedom	4
ks stat(92,5.e-002)	0.14	alpha	5.e-002
p-value	0.539	chi**2(4,5.e-002)	9.49
result	DO NOT REJECT	p-value	0.203
9-10		9-10	
Poisson		Kolmogorov-Smirnov	
lamda =	2.16304	data points	92
Chi Squared		ks stat	8.2e-002
total classes	6	alpha	5.e-002
interval type	equal probable	ks stat(92,5.e-002)	0.14
net bins	5	p-value	0.539
chi**2	5.95	result	DO NOT REJECT
degrees of freedom	4	9-10	
alpha	5.e-002	Poisson	
chi**2(4,5.e-002)	9.49	lamda =	2.16304
p-value	0.203	Chi Squared	
result	DO NOT REJECT	total classes	6
Kolmogorov-Smirnov		interval type	equal probable
data points	92	net bins	5
ks stat	8.2e-002	chi**2	5.95
alpha	5.e-002	degrees of freedom	4
ks stat(92,5.e-002)	0.14	alpha	5.e-002
p-value	0.539	chi**2(4,5.e-002)	9.49
result	DO NOT REJECT	p-value	0.203

MODELO DE SIMULACIÓN BASADO EN AGENTES PARA RESOLVER EL PROBLEMA DE RUTEO EN EL CUIDADO DE LA SALUD DOMICILIARIA

<p style="text-align: center;">10-11</p> <p>Poisson lamda = 1.86957</p> <p>Chi Squared total classes 6 interval type equal probable net bins 5 chi**2 8.41 degrees of freedom 4 alpha 5.e-002 chi**2[4,5.e-002] 9.49 p-value 7.75e-002 result DO NOT REJECT</p> <p>Kolmogorov-Smirnov data points 92 ks stat 7.41e-002 alpha 5.e-002 ks stat[92,5.e-002] 0.14 p-value 0.666 result DO NOT REJECT</p>	<p style="text-align: center;">11-12</p> <p>Poisson lamda = 1.42391</p> <p>Chi Squared total classes 8 interval type equal probable net bins 4 chi**2 2.06 degrees of freedom 3 alpha 5.e-002 chi**2[3,5.e-002] 7.81 p-value 0.559 result DO NOT REJECT</p> <p>Kolmogorov-Smirnov data points 92 ks stat 5.77e-002 alpha 5.e-002 ks stat[92,5.e-002] 0.14 p-value 0.902 result DO NOT REJECT</p>
<p style="text-align: center;">12-13</p> <p>Geometric p = 0.362205</p> <p>Chi Squared total classes 13 interval type equal probable net bins 4 chi**2 6.94 degrees of freedom 3 alpha 5.e-002 chi**2[3,5.e-002] 7.81 p-value 7.37e-002 result DO NOT REJECT</p> <p>Kolmogorov-Smirnov data points 92 ks stat 9.32e-002 alpha 5.e-002 ks stat[92,5.e-002] 0.14 p-value 0.378 result DO NOT REJECT</p>	<p style="text-align: center;">13-14</p> <p>Poisson lamda = 1.3913</p> <p>Chi Squared total classes 6 interval type equal probable net bins 4 chi**2 3.84 degrees of freedom 3 alpha 5.e-002 chi**2[3,5.e-002] 7.81 p-value 0.279 result DO NOT REJECT</p> <p>Kolmogorov-Smirnov data points 92 ks stat 6.65e-002 alpha 5.e-002 ks stat[92,5.e-002] 0.14 p-value 0.786 result DO NOT REJECT</p>
<p style="text-align: center;">14-15</p> <p>Poisson lamda = 1.31522</p> <p>Chi Squared total classes 6 interval type equal probable net bins 4 chi**2 1.64 degrees of freedom 3 alpha 5.e-002 chi**2[3,5.e-002] 7.81 p-value 0.65 result DO NOT REJECT</p> <p>Kolmogorov-Smirnov data points 92 ks stat 4.68e-002 alpha 5.e-002 ks stat[92,5.e-002] 0.14 p-value 0.982 result DO NOT REJECT</p>	<p style="text-align: center;">15-16</p> <p>Poisson lamda = 1.6413</p> <p>Chi Squared total classes 6 interval type equal probable net bins 4 chi**2 11.6 degrees of freedom 3 alpha 5.e-002 chi**2[3,5.e-002] 7.81 p-value 8.82e-003 result REJECT</p> <p>Kolmogorov-Smirnov data points 92 ks stat 7.7e-002 alpha 5.e-002 ks stat[92,5.e-002] 0.14 p-value 0.619 result DO NOT REJECT</p>
<p style="text-align: center;">16-17</p> <p>Poisson lamda = 2.03261</p> <p>Chi Squared total classes 6 interval type equal probable net bins 5 chi**2 2.61 degrees of freedom 4 alpha 5.e-002 chi**2[4,5.e-002] 9.49 p-value 0.625 result DO NOT REJECT</p> <p>Kolmogorov-Smirnov data points 92 ks stat 4.29e-002 alpha 5.e-002 ks stat[92,5.e-002] 0.14 p-value 0.993 result DO NOT REJECT</p>	<p style="text-align: center;">17-18</p> <p>Poisson lamda = 2.08696</p> <p>Chi Squared total classes 6 interval type equal probable net bins 5 chi**2 3.06 degrees of freedom 4 alpha 5.e-002 chi**2[4,5.e-002] 9.49 p-value 0.547 result DO NOT REJECT</p> <p>Kolmogorov-Smirnov data points 92 ks stat 3.9e-002 alpha 5.e-002 ks stat[92,5.e-002] 0.14 p-value 0.998 result DO NOT REJECT</p>

18-19		19-20	
Poisson		Poisson	
lamda =	2.30435	lamda =	2.1087
Chi Squared		Chi Squared	
total classes	6	total classes	6
interval type	equal probable	interval type	equal probable
net bins	5	net bins	5
chi**2	10.	chi**2	4.21
degrees of freedom	4	degrees of freedom	4
alpha	5.e-002	alpha	5.e-002
chi**2[4,5.e-002]	9.49	chi**2[4,5.e-002]	9.49
p-value	4.04e-002	p-value	0.379
result	REJECT	result	DO NOT REJECT
Kolmogorov-Smirnov		Kolmogorov-Smirnov	
data points	92	data points	92
ks stat	8.4e-002	ks stat	6.03e-002
alpha	5.e-002	alpha	5.e-002
ks stat[92,5.e-002]	0.14	ks stat[92,5.e-002]	0.14
p-value	0.508	p-value	0.871
result	DO NOT REJECT	result	DO NOT REJECT
20-21		21-22	
Poisson		Poisson	
lamda =	1.52174	lamda =	0.956522
Chi Squared		Chi Squared	
total classes	6	total classes	4
interval type	equal probable	interval type	equal probable
net bins	4	net bins	3
chi**2	0.388	chi**2	3.84
degrees of freedom	3	degrees of freedom	2
alpha	5.e-002	alpha	5.e-002
chi**2[3,5.e-002]	7.81	chi**2[2,5.e-002]	5.99
p-value	0.943	p-value	0.147
result	DO NOT REJECT	result	DO NOT REJECT
Kolmogorov-Smirnov		Kolmogorov-Smirnov	
data points	92	data points	92
ks stat	1.8e-002	ks stat	5.61e-002
alpha	5.e-002	alpha	5.e-002
ks stat[92,5.e-002]	0.14	ks stat[92,5.e-002]	0.14
p-value	1.	p-value	0.919
result	DO NOT REJECT	result	DO NOT REJECT
22-23		23-24	
Poisson		Poisson	
lamda =	0.554348	lamda =	0.23913
Chi Squared		Chi Squared	
GOODNESS OF FIT: too few intervals for a good chi-squared test		GOODNESS OF FIT: too few intervals for a good chi-squared test	
Kolmogorov-Smirnov		Kolmogorov-Smirnov	
data points	92	data points	92
ks stat	1.25e-002	ks stat	1.56e-002
alpha	5.e-002	alpha	5.e-002
ks stat[92,5.e-002]	0.14	ks stat[92,5.e-002]	0.14
p-value	1.	p-value	1.
result	DO NOT REJECT	result	DO NOT REJECT