

# Método para Manejar el Problema de la Recarga de Trabajo en los Sistemas Multi-Agente

## Caso de estudio: Sistema Multi- Agente de Planificación para la Ubicación de Zonas de Depósito “SMAP-UZD”

Daniel Horfan Álvarez <sup>\*</sup>    Demetrio A. Ovalle Carranza <sup>†</sup>

### Resumen

Los Sistemas Multi-Agente (SMA) permiten la solución de un problema mediante la descomposición de éste en subproblemas, cada uno de los cuales puede ser solucionado por uno o un grupo de agentes en particular. Para optimizar el tiempo de ejecución de tareas en un SMA se requiere que éste presente características especiales, como son: La división del problema en subproblemas de tiempos de ejecución similares; y una programación de tareas para cada agente, de tal forma que puedan efectuarse aproximadamente en forma paralela y no serial, buscando que el tiempo de ejecución total sea el mínimo. Este proyecto propone la elaboración de un método para la operación de los SMA, que permita la optimización del tiempo de ejecución mediante la distribución uniforme de cargas de trabajo dentro del sistema. Para lograrlo es necesario realizar variaciones en las características tanto de los agentes como del sistema, con el fin de considerar el tiempo como una variable indispensable en la sincronización de la ejecución de las tareas. Las características del sistema que se tuvieron en cuenta para la optimización del tiempo de ejecución fueron: la organización, cooperación, coordinación, control y su protocolo de comunicación.

**Palabras clave:** Sistemas Multiagente -SMA-, Agentes Planificadores, Clonación de Agentes, Recargas de Trabajo en SMA.

### Abstract

Multiagent systems allow the problem solution through the decomposition of it into sub problems; each of them can be solved through one or a group of particular agents. For optimizing the execution time of the tasks within a multi agent system - MAS – it is needed the presence of special characteristics, such as: the division of the problem into sub problems with similar execution time; and a task programming for every agent, so it can be approximately done in parallel and not in serial way, seeking that the total execution time will be minimum. If the MAS does not present any of these characteristics, the total use time by the agents to develop the work will be higher. This project proposes the elaboration of a method to operate the MAS, that allows the optimization of the execution time through the uniform distribution of the charge of work within the system. To fulfill this it is necessary to make some variations in the characteristics of the agents as in the system, with the aim of considering the time as an indispensable variable for the synchronization and the execution of the tasks. The characteristics of the system considered for the optimization of the execution time was: the organization, cooperation, coordination, control and its communication protocol.

**Keywords:** Multiagent Systems, Agents Clonation, Load Balancing

---

<sup>\*</sup> Facultad de Ingenierías, Universidad de San Buenaventura – Sede Medellín, Colombia.  
[daniel.horfan@usbmed.edu.co](mailto:daniel.horfan@usbmed.edu.co)

<sup>†</sup> Escuela de Sistemas, Facultad de Minas, Universidad Nacional de Colombia – Sede Medellín, Colombia.  
[dovalle@unalmed.edu.do](mailto:dovalle@unalmed.edu.do)

## 1. Introducción

Este proyecto surgió de la necesidad de desarrollar un Sistema Multi-Agente orientado a apoyar los procesos de análisis raster al interior de un SIG. Dentro del planteamiento de la estructura organizacional del sistema se encontró que las características del mismo conllevaban a una acumulación de trabajo en determinados tipos de agentes; de ahí que este proyecto esté orientado a cómo manejar el problema de la sobrecarga de trabajo en los agentes, dotándolos de herramientas propias, utilizando un método apropiado de planificación y creando una estructura organizacional que evite al máximo este problema.

En este trabajo se propone un método para Sistemas Multiagentes que permita la distribución de cargas de trabajo y la optimización del tiempo de ejecución, mediante la inclusión de características especiales a sus agentes, a la estructura organizacional y a su protocolo de comunicación. Esto se logró mediante las siguientes adaptaciones al sistema:

- Se definieron nuevas características a los agentes de software para afrontar el problema de la recarga de trabajo, como son: la posibilidad de subdividir una tarea en subtareas y la inclusión de un módulo de clonación, que permite la distribución de las tareas de un agente entre varios, localizados en la misma máquina o en otra estación comunicada por red.
- Se incluyó dentro del sistema un agente encargado de la planificación del trabajo y la optimización del tiempo de ejecución. Este agente cuenta con algoritmos y reglas que le permiten garantizar la realización del trabajo en un tiempo determinado, basándose en las experiencias de los agentes de ejecución en lo relacionado con los recursos del sistema utilizados en corridas anteriores (uso del procesador, memoria empleada, etc).
- Se analizaron y seleccionaron las características del Sistema para encontrar la arquitectura que mejor se ajustara a las adaptaciones anteriormente incluidas. Como resultado de este análisis se encontró que la combinación que mejor se acomodaba al sistema es: una Estructura Organizacional Jerárquica Formal, utilizando Cooperación por Delegación, con una Coordinación Global cuyo control lo realiza un agente planificador.
- Para el funcionamiento adecuado del modelo, se realizó una adaptación al protocolo de comunicación propuesto en KQML (Knowledge Query and Manipulation Language), dividiendo el Nivel de Contenido en dos subniveles encargados de comunicar el qué y el cómo se debe realizar una tarea.

Para validar el modelo propuesto se implementó un prototipo de un Sistema Multiagente de Apoyo a los Análisis Raster en Sistemas de Información Geográfica. Este fue desarrollado en Visual Basic utilizando Clips como motor de inferencia para los agentes que funcionan como sistemas expertos.

En los numerales siguientes se presentan los aspectos teóricos generales acerca de los SMA, las características del método propuesto, el diseño e implementación del prototipo y los resultados obtenidos con el mismo.

## 2. Agentes y Sistemas Multiagentes

### 2.1 Agentes

En los trabajos relacionados con agentes se ha planteado una variedad de definiciones, que a partir de sus elementos comunes permiten definir un agente como un sistema computacional que opera de forma autónoma, con capacidad de reaccionar dinámicamente a estímulos externos y comunicarse entre sí y que posee conocimientos e intencionalidad para resolver problemas específicos. [3][10][7].

A los agentes se asocian características como: Intencionalidad, entendida como la declaración explícita de metas y medios para llegar a ellas; racionalidad, que es la capacidad de evaluar y seleccionar acciones a ejecutar; compromiso, consistente en la planificación de las tareas por medio de la coordinación y negociación entre agentes; adaptabilidad, referida al control de sus aptitudes y comportamientos de acuerdo con las funciones que ejecuta. Así mismo, los agentes pueden clasificarse desde diferentes puntos de vista en estáticos o móviles según su movilidad; en deliberativos y reactivos de acuerdo con su capacidad de reacción; y en autónomos o no, si operan sin necesidad de intervención humana. [5][11].

Los agentes de mayor interés al hablar de inteligencia artificial son los autónomos, pues presentan características que los acercan al comportamiento humano, tales como: racionalidad, habilidad social, cooperación, capacidad de aprendizaje, reactividad y proactividad.

Aquellos agentes que combinan características propias de varias de las clasificaciones enunciadas anteriormente son denominados Híbridos. Éstos se utilizan en aquellas aplicaciones en donde se considera que trae más beneficios tener un sólo agente con varias propiedades, en lugar de varios con cada una de ellas. Por ejemplo, se puede diseñar un agente que posea al mismo tiempo las características de deliberativo y de reactivo.

## 2.2 Sistemas Multi-Agente (SMA)

Desde que surgió la Inteligencia Artificial Distribuida (IAD), se han abordado los temas relacionados con el estudio de los modelos y del comportamiento de varios agentes que cooperan entre sí para la resolución de un problema o desarrollo de una tarea. En particular, el área de los SMA estudia el comportamiento y la interacción de las entidades que integran el sistema. En los SMA, las entidades inteligentes que lo conforman se denominan agentes. [8].

Dentro de las características principales de un SMA se tienen: la *no existencia de benevolencia*, lo cual hace a los agentes menos vulnerables a desviaciones del objetivo principal por la influencias externas; la *obtención de Múltiples metas*, por la capacidad de resolver varios problemas e inclusive enfrentar el problema de varias maneras; la *Autonomía* pues los agentes que conforman el SMA deben tener autonomía que les permita seleccionar la mejor manera para cumplir con sus objetivos y la *Heterogeneidad*, ya que los agentes que conforman un SMA tienen características muy particulares, lo cual les admite intervenir en el problema de diferentes maneras. Esta heterogeneidad hace necesaria la existencia de un sistema de comunicación que permita la interacción entre los diferentes agentes.

Dentro de la Arquitectura de los sistemas multiagentes existen ciertas características que definen el funcionamiento del sistema:

### 2.2.1 Estructura Organizacional

La estructura organizacional está relacionada con la estructura de las componentes funcionales del sistema, sus características, sus responsabilidades, sus necesidades y de la forma como realizan sus comunicaciones. En general, se pueden distinguir cuatro tipos de configuraciones:

- **Estructura Centralizada:** Un agente controla la interacción entre los otros agentes pues posee ese papel.
- **Estructura Horizontal:** Todos los agentes están en el mismo nivel, por lo tanto no hay agentes controladores.
- **Estructura Jerárquica:** Existen diferentes niveles de agentes que trabajan diferentes "niveles de abstracción". Entre los agentes de un mismo nivel, existe una estructura horizontal. Las estructuras jerárquicas se pueden dar de dos formas: Formal. cuando un agente de menor nivel solo puede recibir ordenes del agente de su nivel inmediatamente superior e Informal. cuando puede recibir ordenes de cualquier agente de un nivel superior al suyo.
- **Estructura "ad hoc":** Es una mezcla de las anteriores y eso le da una característica de dinamismo. Los diferentes grupos de agentes se ajustan entre ellos.

### 2.2.2 Cooperación

La cooperación se refiere a la forma como los agentes trabajan conjuntamente para lograr un objetivo global, ésta depende de la configuración organizacional del sistema. Existen varios modelos para lograr esto:

- **Cooperación compartiendo tareas y resultados:** Con base en los resultados generados por los demás agentes, un agente realiza sus propias tareas.
- **Cooperación por delegación:** El agente supervisor divide una tarea en subtareas, las asigna a sus agentes para su ejecución, y por ultimo, las integra para hallar la solución global.
- **Cooperación por ofrecimiento:** El agente supervisor divide una tarea en subtareas y las ubica en una lista esperando que los agentes del sistema se ofrezcan a realizarlas dependiendo de sus habilidades. El supervisor elige entre los ofrecimientos y las distribuye.[6]

### 2.2.3 Coordinación

La coordinación entre un grupo de agentes les permite considerar todas las tareas a realizar por medio de una planificación de acciones para la resolución de tareas. Esta planificación lleva a que se conozca con anticipación el comportamiento de los agentes y por tanto, a través de la coordinación, permitir que un grupo de agentes analice las tareas por realizar y se coordinen para ejecutarlas. Existen varios modelos de coordinación de acciones entre agentes:

- **Coordinación Global:** Cuando el SMA determina y planifica globalmente las acciones de los diferentes agentes.
- **Coordinación Individual:** Cuando el SMA da autonomía a los agentes, es decir cada agente decide qué hacer y resuelve localmente los conflictos que detecte con otros agentes.

### 2.2.4 Control

Es un instrumento que provee apoyo en la implementación de mecanismos de coordinación. El control puede ser considerado desde el punto de vista local y global. Se debe mantener un balance entre el control global (se relaciona con el control que se puede tomar basándose en la información producida por todos los agentes del sistema) y local (solamente con base en datos locales) pues si se da mucho control global se puede aumentar considerablemente el tiempo de cómputo, por el cambio dinámico de la información producida por los agentes y si se desborda el control local, se pierde eficiencia en la cooperación pues se pueden realizar tareas no deseables.

## 2.3 Lenguaje de comunicación KQML [2]

La comunicación en los sistemas multiagentes es primordial, ya que es el medio mediante el cual los agentes comparten conocimiento y se sincronizan para llevar a cabo sus tareas conjuntamente. KQML se planteó en función de un modelo de interacción dinámica entre agentes inteligentes llamado KBS (Knowledge Based System). Este modelo permite evitar las limitaciones que imponían los modelos distribuidos, y a la vez hacer más fácil la interacción entre verdaderos agentes inteligentes. El modelo más simple es el cliente-servidor, en el cual uno de los agentes actúa como cliente formulando una petición a otro agente, el cual actúa como servidor proporcionando la respuesta.

La comunicación entre agentes utilizando KQML puede ser sincrónica o asincrónica. La primera de ellas se refiere a que el mensaje no se envíe de forma completa sino fraccionada, estableciéndose una comunicación entre los dos agentes de forma que el cliente vaya pidiendo al servidor sucesivamente las respuestas; la asincrónica consiste en que el cliente se suscribe a un determinado servidor que, de manera asincrónica, envía la respuesta.

### 2.3.1 Estructura KQML

Dentro de la estructura de comunicación de KQML se distinguen los siguientes niveles estructurados de forma similar al modelo OSI:

- **Nivel de mensaje:** Constituye el núcleo del mensaje y es el que determina la clase de interacciones que puede tener un agente con otro<sup>7</sup>.
- **Nivel de contenido:** Constituye el conocimiento o petición que se desee comunicar en un lenguaje determinado por ejemplo una pregunta, una respuesta, una orden, u otro requerimiento.
- **Nivel de transporte o comunicación:** Contiene los parámetros básicos de la comunicación como la identidad del emisor y el receptor.

## 3. Método de Distribución de Cargas de Trabajo en SMA [4]

Los sistemas multiagentes pueden tener comportamientos poco eficientes en algunos de los trabajos adelantados, éstos se puede asociar con una mala planificación o a la falta de recursos computacionales suficientes. El método propuesto trata de dar solución a uno de los problemas comunes de los sistemas multiagentes, como es la recarga de trabajo en uno o varios agentes, que pueden retardar la ejecución total del trabajo. La solución a este problema se basa en atacar los puntos susceptibles de optimizar en el sistema y en lo posible redistribuir los cálculos que demandan gran cantidad de recursos en varias estaciones de trabajo, basados en la técnica de clonación de agentes.

Para la ubicación de los aspectos a optimizar en el sistema basta hacer una analogía con la vida real, identificando los pasos que son necesarios en una organización para la resolución de un problema y luego tratar de llevarlos al sistema multiagente en forma de base conocimiento, es decir, reglas que le permitan optimizar su funcionamiento.

Dentro de una organización sea de agentes computacionales o personas, los aspectos vulnerables durante la ejecución de un trabajo son:

### 3.1 Planificación de Agentes

Dentro del proceso de planificación se incluyeron factores que pueden ser determinantes en el tiempo de ejecución de los trabajos como son los recursos computacionales, la disponibilidad de los agentes y el tiempo de ejecución empleado por cada agente. Este conocimiento lo obtiene el sistema de la información histórica relacionada con los tiempos de ejecución y la forma de solucionar un problema por parte de los otros agentes.

#### 3.1.1 Descripción del Proceso de Planificación

En la figura 1 se presenta un diagrama con el proceso de planificación de un SMA que reúne los elementos anteriormente descritos, donde: Inicialmente se recibe la Información del Problema, se subdivide en tareas, se encuentra qué agente puede realizarlas, se negocia con éste las características de la ejecución, se crea el cronograma de trabajo, se confirman las características de la ejecución y se asignan a los agentes para que se lleven a cabo el trabajo. Luego, durante la ejecución del trabajo, el agente planificador realizará auditorías periódicas con el fin de verificar la correcta ejecución del sistema.

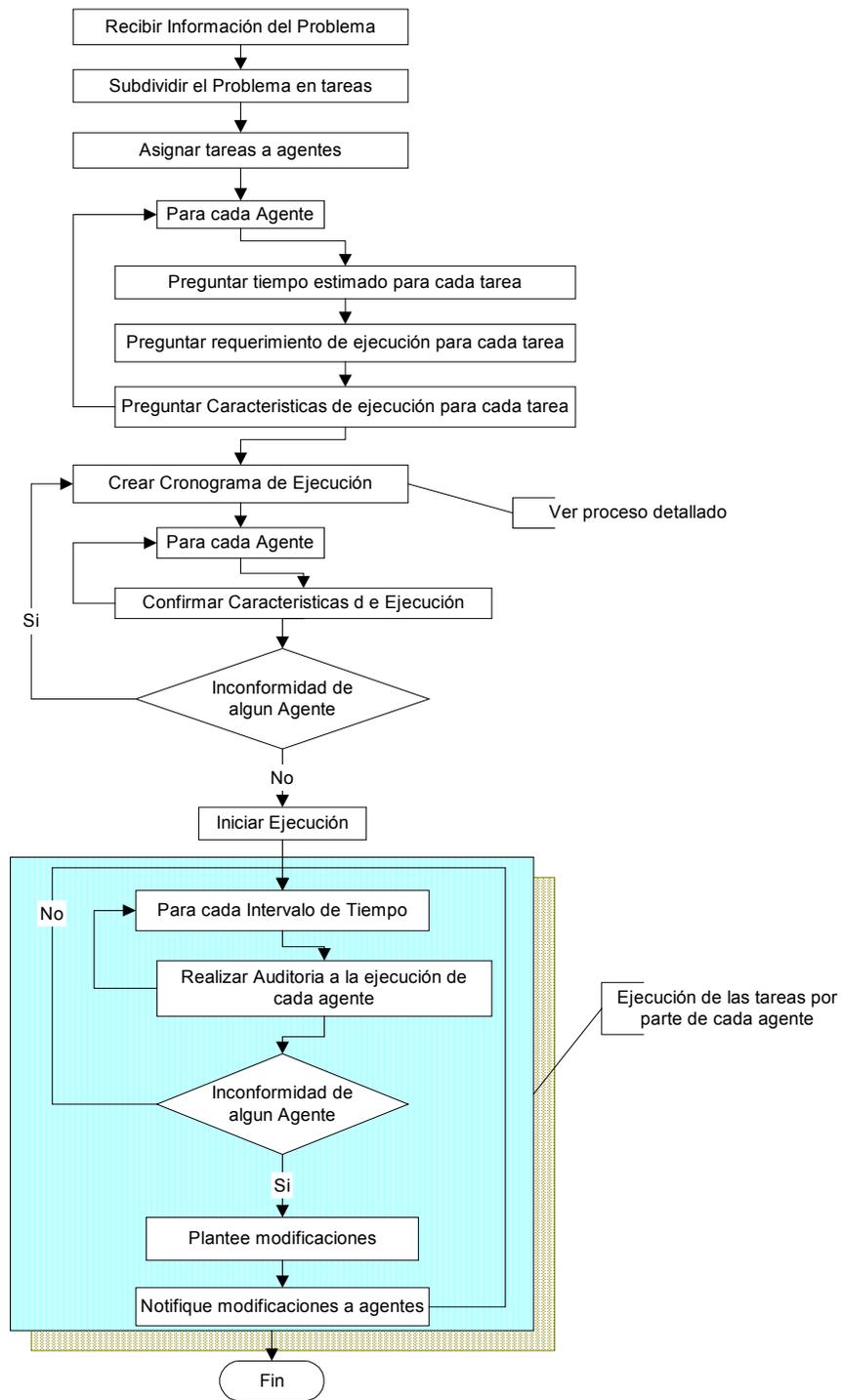


Figura 1. Proceso de planificación de un SMA

### 3.1.2 Algoritmo para generación del cronograma de ejecución

La generación del cronograma es la principal actividad en el proceso de planificación ya que en ésta se determina la organización lógica de ejecución de las tareas. En la figura 3 se presenta el diagrama de flujo que representa el algoritmo para la generación del cronograma de ejecución.

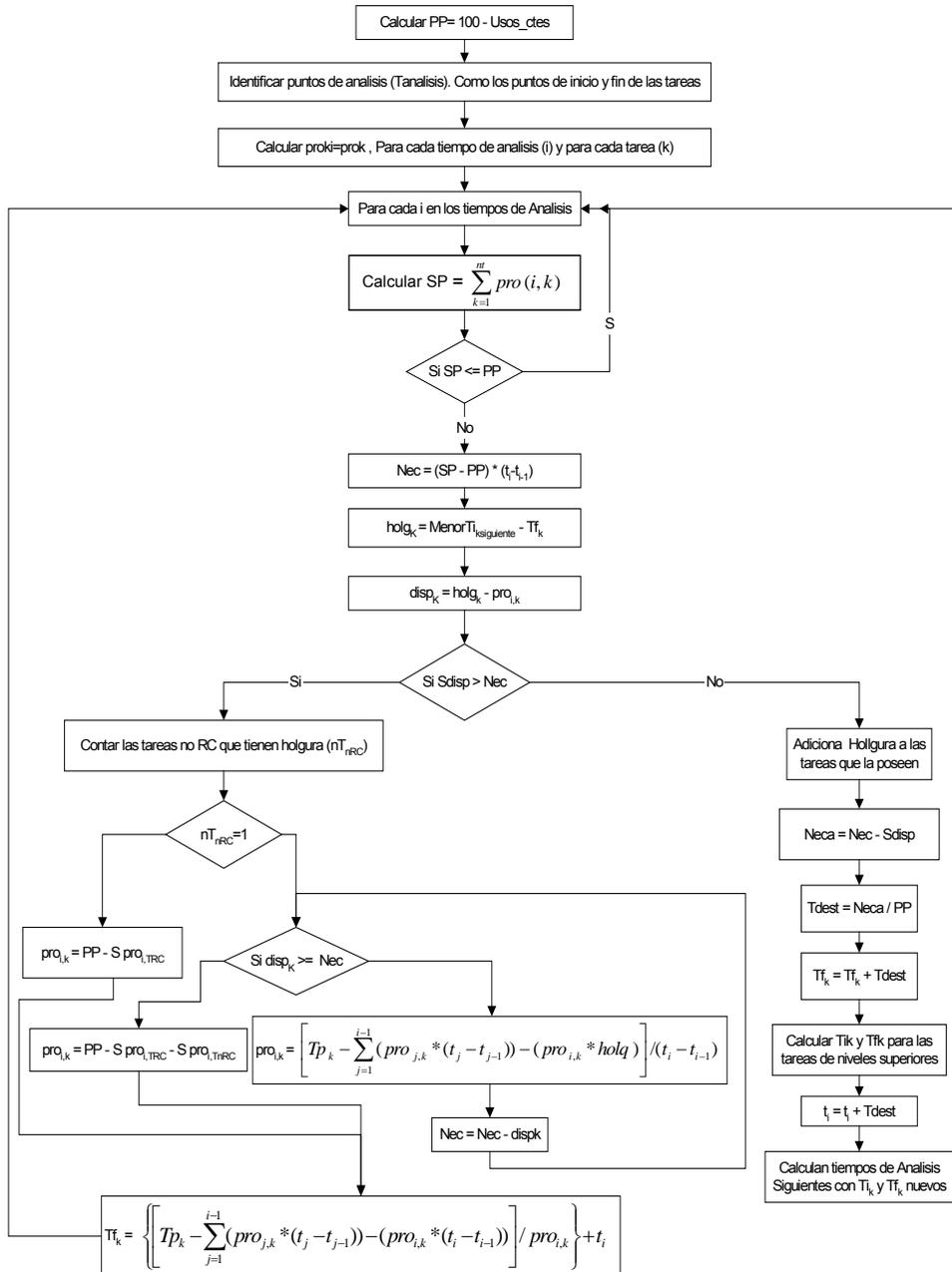


Figura 2. Proceso de cálculo del cronograma de ejecución

Como paso inicial del proceso se deben calcular los datos iniciales y guardarlos en una estructura como la presentada en la siguiente tabla donde se muestran 6 tareas a forma de ejemplo para realizar su cronograma.

Tabla 1. Organización de las Tareas según sus prerequisites

Id	Tareas	Niv	Prec	Ti	Tf	dT	RC	Pro	Tp	Clon	Procesador(horas)						
											1	2	3	4	5	6	7
1	Tarea1	1		0.0	1.5	1.5	X	80	120	0	80						
2	Tarea2	2	1	1.5	3.0	1.5		50	75	50		50					
3	Tarea3	2	1	1.5	4.2	2.7	X	80	216	0		80					
4	Tarea4	1		0.0	2.7	2.7		70	189	0	70						
5	Tarea5	3	2,3	4.2	5.5	1.3	X	70	91	0				70			
6	Tarea6	4	4,5	5.5	7.0	1.5	X	95	143	0							95

Donde:

- Id: Es un número asignado a cada tarea
- Tareas: Descripción de la tarea
- Niv: Nivel de precedencia al que pertenece cada tarea, donde 1 se le asigna a tareas que no tienen precedencia, 2 a las que dependen de las 1 y así sucesivamente.
- Prec: Precedencias o prerequisites de cada tarea
- Ti : Tiempo de inicio de la tarea
- Tf : Tiempo de terminación de la tarea
- DT : Duración de la tarea
- RC : Señalador que indica que una tarea se encuentra en la Ruta Crítica
- Pro : Uso del procesador por unidad de tiempo durante la ejecución de la tarea
- Tp : Uso total del procesador en la tarea
- Clon: Uso del procesador en otro PC donde está en ejecución un clon

En la tabla anterior se observa que para un intervalo de tiempo el uso del procesador supera los recursos posibles, se puede deducir que el PC no puede desarrollar las tareas en el tiempo especificado y que será necesario modificar el cronograma.

Para balancear la carga del procesador primero se debe calcular la cantidad de procesador disponible y compararlo contra la necesidad de procesador en un intervalo de tiempo dado, en caso de ser mayor la necesidad que la disponibilidad se requiere entonces conocer si las tareas que tienen holgura pueden cambiar su utilización del procesador para cumplir con la disponibilidad, si es así basta con modificar los tiempos finales y uso del procesador para esas tareas; en caso contrario es necesario mover las fechas de procesamiento alterando el tiempo final de ejecución.

En las tablas 2 y 3 se presentan los estados de las tareas en el cronograma antes y después del balanceo de cargas para el ejemplo mostrado en la tabla 1

Tabla 2. Usos del procesador para cada tiempo de Análisis Antes de realizar el balance de cargas

k	Tareas	I						Pro
		1.5	2.7	3.0	4.2	5.5	7.0	
1	Tarea1	80	0	0	0	0	0	80
2	Tarea2	0	50	50	0	0	0	50
3	Tarea3	0	80	80	80	0	0	80
4	Tarea4	70	70	0	0	0	0	70
5	Tarea5	0	0	0	0	70	0	70
6	Tarea6	0	0	0	0	0	95	95

Tabla 3. Usos del procesador para cada tiempo de Análisis después de realizar el balance de cargas

k	Tareas	I				
		1.5	3.0	5.44	7.27	8.77
1	Tarea1	80	0	0	0	0
2	Tarea2	0	10	24.53		0
3	Tarea3	0	80	39.25		0
4	Tarea4	15	5	31.21	45.22	0
5	Tarea5	0	0	0	49.77	0
6	Tarea6	0	0	0	0	95

Cabe señalar que en la fase de “Iniciar ejecución” del proceso de planificación, al momento de asignar las tareas a los diferentes agentes se debe tener en cuenta las siguientes consideraciones:

- Tareas cuyo  $pro_{i,k}$  fue siempre menor que su  $pro_k$  son tareas de baja prioridad
- Tareas que pertenecen a la ruta crítica deben ser ejecutadas con alta prioridad
- El resto de tareas deben ser ejecutadas con prioridad media o normal

### 3.2 Técnicas o habilidades propias de cada agente

Otro aspecto fundamental en un SMA son las habilidades propias del agente para llevar a cabo su labor, ya que éstas son las que le permitirán cumplir con los tiempos estimados durante la etapa de planificación. Además de las habilidades mencionadas anteriormente se asignarán a los agentes la capacidad de clonación y de subdividir su trabajo y redistribuirlo.

#### 3.2.1 Clonación de Agentes

La clonación en los sistemas multiagentes consiste en crear o reproducir agentes con las mismas características y capacidad [9]. Lo primero que se debe tener en cuenta para el proceso de clonación es la necesidad del clon, ésta se determina mediante una serie de sensores que verifican el avance real versus el planeado, que cuando detectan un atraso considerable inician el proceso de clonación, el cual consta de los siguientes pasos (Ver figura 3).

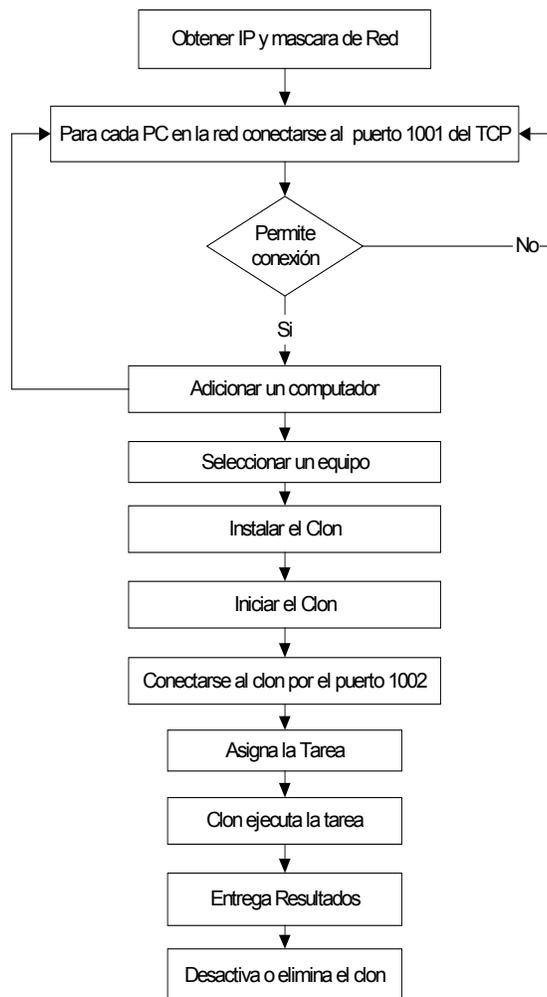


Figura 3. Diagrama de flujo del proceso de clonación de agentes

### 3.2.2 Redistribución de cálculos que demandan gran cantidad de recursos

Una buena estrategia para optimizar la ejecución de una tarea desarrollada por un agente es la división de ésta en subtareas, para ser asignadas a clones que las ejecuten, esto implica tener en cuenta las características de la tarea para identificar la posibilidad de la desagregación.

### 3.3 Protocolos de Comunicación

El protocolo de comunicación propuesto no difiere mucho del expuesto en KQML, se diferencia en que se discriminan dos partes en el nivel de contenido: el **Qué** y el **Cómo**. Donde el **Qué** se utiliza para expresar al agente destinatario la actividad que debe realizar y el **Cómo** para indicar la forma como debe llevar a cabo su labor. (Ver figura 4)



Nivel	Comunicación		Mensaje	Contenido	
	Origen	Destino	Realizar tarea	Qué	Cómo
Ejemplo				Servicio y parámetros necesarios	Condiciones para la ejecución

Figura 4. Niveles de Abstracción del Protocolo de Comunicación KQML modificado

### 3.4 Modelos Multiagentes

Como se pudo observar en el capítulo anterior, existen varias formas de diseñar el modelo de un Sistema Multiagente, combinando diferentes características de acuerdo con las características de estructura organizacional, cooperación y coordinación. Cada una de éstas puede favorecer o no, el desempeño del sistema.

Como resultado de este análisis se encontró que la combinación que mejor se acomodaba al sistema es: una Estructura Organizacional Jerárquica Formal, utilizando Cooperación por Delegación, con una Coordinación Global cuyo control lo realiza un agente planificador.

## 4. Diseño, Implementación y Validación del Sistema Multiagente de Planificación (SMAP) [4]

El prototipo SMAP comprende el diseño e implementación de un sistema multiagente desarrollado como herramienta de apoyo en la determinación de la ubicación de zonas de depósitos de materiales de construcción, usando la superposición de imágenes de diferentes temas o mapas que den un primer diagnóstico de su ubicación.

El SMAP propuesto denominado, “Sistema Multiagente de Planificación para la Ubicación de Zonas de Deposito” (SMAP-UZD), servirá como piloto para analizar las mejoras en el desempeño de un SMA utilizando lo expuesto en el capítulo anterior.

### 4.1 Análisis y diseño del SMAP-UZD

#### 4.1.1 Modelo del agente

Un agente dentro del SMAP es una aplicación autónoma representada por varios componentes que unidos le ofrecen un comportamiento y una utilidad particular. Dentro de la estructura interna de cada agente se pueden definir tres partes principales:

- **Componente de Ejecución:** Es la parte del agente encargada de llevar a cabo las tareas asignadas a él, mediante una serie de funciones o subprogramas que el agente invoca de acuerdo con su conocimiento y experiencia. Está normalmente conformada por: Utilidades, Base de datos y su interfaz.
- **Componente de Funcionamiento:** Se trata de la parte más importante del agente, pues en ésta se encuentran las utilidades que le dan su autonomía y comportamiento. Está compuesta por: Planificación y control, Clonación, Base de Conocimiento Funcional y su Interfaz de Monitoreo
- **Componente de Comunicación:** Son las utilidades que le permiten interactuar con los demás agentes del sistema. Se puede decir que se trata de la boca y los oídos del agente.

#### 4.1.2 Estructura del SMAP-UZD

El SMAP-UZD utiliza una estructura organizacional jerárquica que comprende tres niveles de autoridad de los agentes (Ver Figura 5), utiliza una cooperación compartiendo tareas y resultados entre los agentes de procesos y sus clones y una cooperación por delegación en la interacción del agente planificador y los agentes de procesos y resultados. La coordinación y el control del sistema están a cargo del agente planificador.

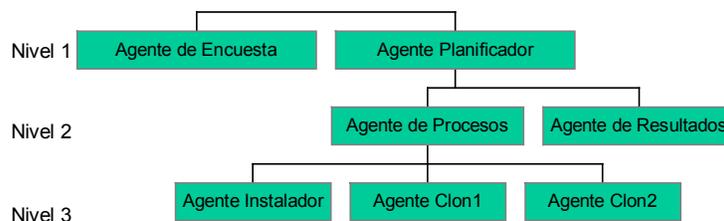


Figura 5. Estructura Organizacional Jerárquica del SMAP-UZD

## 5. Evaluación del Método

Para la evaluación del método planteado se realizaron cinco corridas en el SMAP-UZD, todas ellas con iguales características en cuanto a la cantidad de trabajo total, pero modificando algunas condiciones internas del sistema adicionadas como parte del método propuesto, como son: clonación, planificación y protocolo de comunicación.

### 5.1 Resultados

Los resultados obtenidos se pueden observar en la siguiente tabla:

Tabla 4. Resultados Obtenidos de los cinco escenarios considerados

Ejecución de casos	Tiempo de ejecución (min:seg)	Tiempo de ejecución (seg)	Uso promedio del Procesador	Tiempo real – planeado (min:seg)
A (completo)	14:03	843	98.9%	0:32
B (sin clonación)	14:21	861	97.5%	0:39
C (sin planeación)	15:17	917	98.0%	–
D (sin clonación, ni planeación)	15:35	935	97.8%	–

Adicionalmente como parte de los resultados se muestran las siguientes figuras de uso del procesador obtenidos en las ejecuciones anteriores:

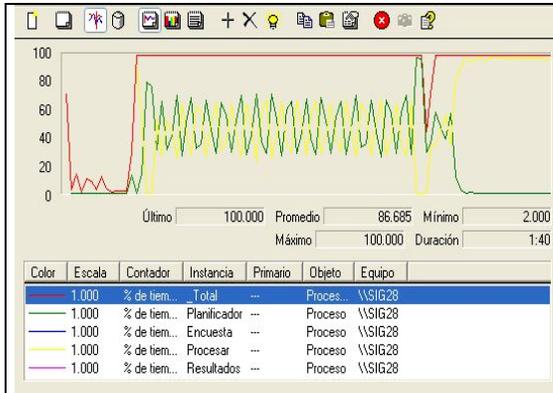


Figura 6. Estado del procesador en el proceso de planificación (Caso A)

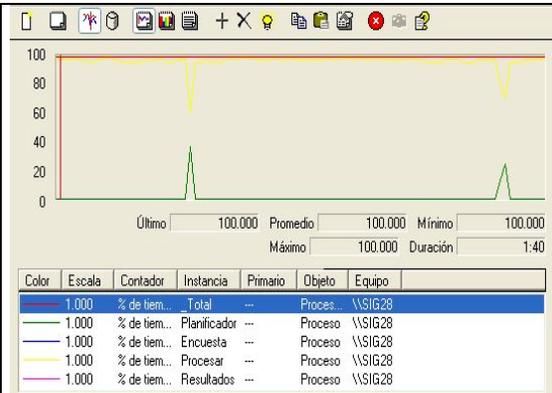


Figura 7. Estado del procesador en el procesamiento de los mapas después de realizado el proceso de planificación (Caso A)

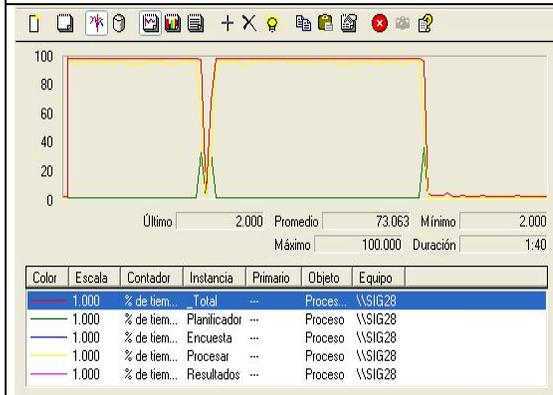


Figura 8. Estado del Procesador al final del cálculo de los mapas, sin utilizar el proceso de clonación (Caso B)

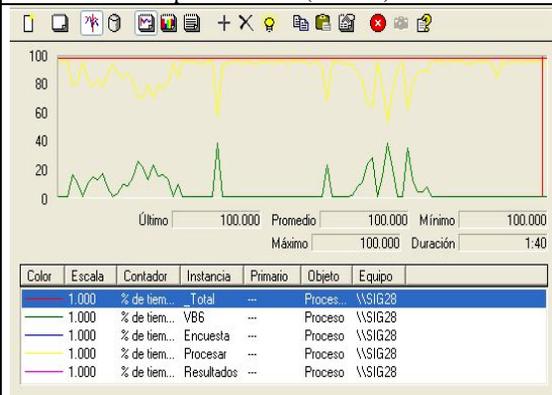


Figura 9. Estado del procesador durante el proceso ejecución de tareas sin planificación (Caso C)

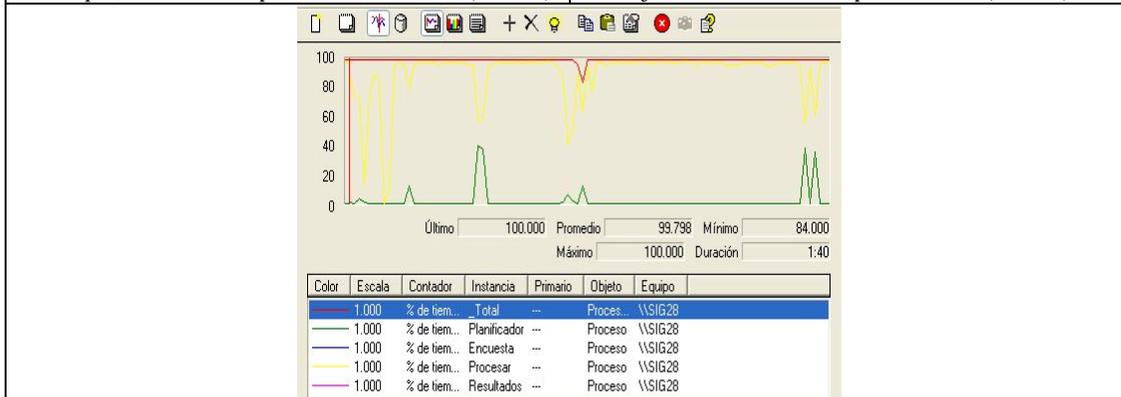


Figura 10. Estado del procesador durante el procesamiento de los mapas, sin utilizar planificación ni clonación (Caso D)

En la Figura 6, se observa la interacción entre el Agente Planificador y el Agente de Procesos, durante el proceso de generación del cronograma de trabajo, en el cual el agente planificador pregunta al de procesos los requerimientos y tiempos estimados para realizar cada una de las labores que tendrá a cargo. Con esta información el agente planificador genera el cronograma inicial y luego lo reajusta teniendo en cuenta el procesador disponible en la estación de trabajo.

En la Figura 7, se observa la interacción entre el agente Planificador y el agente de Procesos, durante el proceso de cálculo de cada uno de los mapas. Las conversaciones entre los agentes sólo se presentan a la entrega y asignación de los trabajos, en todo el tiempo de procesamiento se observa un uso total del procesador por parte del agente de procesos.

En la Figura 8, se observa la interacción entre el agente Planificador y el agente de Procesos, al finalizar el proceso de cálculo de los mapas. Se aprecia un bajón en la línea roja que se presenta cuando se realizan entregas y asignación de trabajos. Si se tuviera un clon en ejecución este espacio de procesador sería utilizado por él en sus procesamientos.

En la Figura 9, se observa la interacción entre el agente Planificador y el Agente de procesos, durante el cálculo de los mapas sin haber realizado un proceso previo de planificación. En este caso se presenta una participación mayor del agente planificador durante el procesamiento de los mapas.

En la Figura 10, se muestra la interacción entre el agente Planificador y el Agente de Procesos durante el cálculo de los mapas, sin haber realizado un proceso previo de planificación y sin tener clones que utilicen el procesador remanente. En la figura se observa cómo el agente de procesos permanece bastante tiempo sin utilizar la totalidad del procesador y se observan utilizaciones hasta del 84% del procesador, lo que implica retrasos en el tiempo de ejecución total.

## **5.2 Análisis de resultados**

### **5.2.1 Comparación Caso A y Caso B**

En el Caso B (sin clonación), se observan períodos de tiempo en los que el sistema no aprovecha toda la disponibilidad del procesador, lo que no sucede en el Caso A (con todas las utilidades), en el cual, estos remanentes son utilizados por los clones para adelantar tareas.

Sin embargo, la diferencia de tiempo entre el Caso A y Caso B es poca, ya que los clones son creados en la misma máquina; si éstos fueran generados en otras estaciones de la red el ahorro en tiempo sería mayor.

### **5.2.2 Comparación Caso A y Caso C**

En el caso planificado se observa que existe un período inicial de iteración entre los agentes y luego toda la prioridad en el procesador le es entregada al agente de Procesos (ver Figura 6); esto contribuye a un máximo aprovechamiento del recurso disponible y así sólo se comunica con el agente planificador para entregar el resultado de su trabajo y recibir nuevas asignaciones de tareas. Por otro lado, se observa en el caso no planificado (ver Figura 9) que la iteración entre los agentes es más activa durante todo el tiempo, pues el Agente Planificador envía los trabajos al de Procesos sin tener en cuenta cómo es su prioridad y ubicación dentro del proceso total, lo cual genera inconsistencias en el agente de procesos que le hacen abortar la tarea y comunicar la imposibilidad de realizar la misma al Agente Planificador, el cual debe intentar asignar una nueva tarea. Este proceso crea un ciclo que puede afectar negativamente el desempeño del sistema.

El retraso de un minuto y catorce segundos presentado por el Caso C (sin planeación) se debió al desorden inicial en las tareas a ejecutar sin planificación; si las tareas inicialmente se encontraran en un orden adecuado, es posible que el sistema sin planificación demore menos

tiempo que el planificado, sin embargo, en la mayoría de los casos esto no ocurre, y este desorden causa demoras variables que perjudican los tiempos de ejecución totales del sistema.

Otra ventaja del sistema planificado es que éste le notifica al usuario el tiempo que gastará en la ejecución total del trabajo, lo cual es una ventaja que no presenta el sistema no planificado. Con respecto al ítem “**Tiempo real –Tiempo planeado**” presentado en la Tabla 1, se observa que a medida que se realizan más corridas en el sistema éste disminuye, debido al mayor conocimiento que va adquiriendo el agente de procesos.

### **5.2.3 Comparación Caso A y Caso D**

Esta comparación es la mejor muestra de los beneficios de emplear el método propuesto. Como se puede apreciar en la Tabla 4, el tiempo utilizado en el Caso D (sin clonación, ni planeación) es el mayor de todas las pruebas realizadas, ya que tiene las desventajas expuestas en los numerales 5.2.1 y 5.2.2.

## **6. Conclusiones y Recomendaciones**

El método propuesto para manejar en forma óptima el problema de la recarga de trabajo tiene su principal aplicación en Sistemas Multiagentes (SMA) que demanden gran cantidad de recursos computacionales para sus procesamientos. Dicho método ofrece una alternativa para la optimización de SMA, logrando una reducción en los tiempos de ejecución de tareas y un mayor aprovechamiento de los recursos disponibles.

La utilización de los algoritmos propuestos en esta tesis para que la planificación considere variables externas al sistema como disponibilidad de procesador y de memoria, ofrecen una ventaja muy útil al usuario, ya que lo mantienen informado del tiempo que empleará el sistema para ejecutar el trabajo. Además, hace posible la entrega de los trabajos a los agentes de ejecución en un orden lógico, lo que evita retrasos asociados con inconsistencias por prerequisites en las tareas.

Por otro lado, se concluye que la posibilidad de realizar clonación contribuye a la optimización del sistema, incluso cuando se crean clones dentro de la misma máquina, ya que ellos aprovechan los remanentes de los recursos durante todas las etapas de la ejecución. Sin embargo, la mayor contribución se logra cuando los clones se crean en otras estaciones de trabajo, pues amplían la cantidad de recursos disponibles para ejecutar las tareas.

En cuanto al protocolo de comunicación, su aporte se refleja en la claridad y facilidad para la implementación del módulo de comunicación entre los agentes, dedicándose cada componente a una labor específica e independiente de las otras.

Del análisis de las características del Sistema se concluyó que la combinación que mejor se acomodaba a un SMA de planificación que utilice el método propuesto son: una Estructura Organizacional Jerárquica Formal utilizando Cooperación por Delegación, con una Coordinación Global cuyo control lo realiza un agente planificador.

Se recomienda que para futuras investigaciones en éste tema se analicen otros factores externos como el uso de la memoria. Igualmente sería recomendable combinar la planificación propuesta con la Planificación Global Parcial Generalizada (GPGP) propuesta por Carrillo et al, 1999, como solución al problema de Coordinación Distribuida en entornos Multiagente.

## Referencias

- [1] A. Carrillo, A. Quintero. GPGP (Planeación Global Parcial) Como Solución al Problema de Coordinación Distribuida. *Universidad de los Andes, Departamento de Ingeniería de Sistema., Bogotá.* (<http://agamenon.uniandes.edu.co/~revista/memocoor.txt>), 1999.
- [2] T. Finin, Y. Labrou, J. Mayfield. KQML as an agent communication language. *Reporte Técnico Departamento de Ciencias Computacionales. Universidad de Mayfield Baltimore Country. Baltimore, MD.* <http://www.cs.umbc.edu/agents/introduction/kqmlacl.ps>., 1997.
- [3] B. Hayes-Roth, B. An Architecture for Adaptive Intelligent System. *Artificial Intelligence Review: Special Issue on Agents and Interactivity*, 72, 329-365. 1995.
- [4] D. Horfan. Método para manejar el problema de la recarga de trabajo en Sistemas Multiagentes. *Tesis de Maestría. Universidad Nacional de Colombia – Sede Medellín*, 2003.
- [5] H. Nwana. Software Agents: And Overview. *Intelligent Systems Research. Advanced Applications & Technology Department.* Sep. 1996.
- [6] S. Labidi, y W. Lejoudad. De l'Intelligence Artificielle Distribuée aux Systèmes Multi-Agents. *Revue d'Intelligence Artificielle*, 1993.
- [7] P. Maes. Artificial life Meets Entertainment: Life like Autonomous Agents, *Communications of the ACM*, 38,11, 108- 114. 1995.
- [8] A. Quintero, S. Rueda, M. Ucross. Teoría de Agentes y Sistemas Multiagente: Integración de conceptos básicos. *Universidad de los Andes, Departamento de Ingeniería de Sistema., Bogotá*, 1995.
- [9] S. Onn, K. Sycara; Prasad Chalasani; Somesh Jha. Increasing resource utilization and task performance by agent cloning. *Intelligent Agents V. Agent Theories, Architectures and Languages*. 1998.
- [10] Smith, S.C., A. Cypher and J. Spohrer , "KidSim: Programming Agents Without a Programming Language," *Communications of the ACM*, 37, 7, 55- 67.1994.
- [11] Springer Verlag. Una taxonomía para agentes autónomos. *Proceedings of the third international workshop on agent theories, architectures and languages*, 1996.