

Arquitectura basada en agentes inteligentes para la gestión de configuración de red

Martín Monroy Ríos¹, Edelberto Reyes Torregroza² & María Macareno Mojica²

¹ Martín Monroy Ríos, Magíster en Ciencias Computacionales, candidato a Doctor en Ingeniería Telemática. Profesor Universidad de Cartagena, Correo electrónico: mmonroyr@unicartagena.edu.co

² Edelberto Reyes Torregroza, Estudiante del Programa de Ingeniería de Sistemas, Correo electrónico: enriqueviiiireyes@gmail.com

² María Macareno Mojica, Estudiante del Programa de Ingeniería de Sistemas, Correo electrónico: mayomac@gmail.com

RESUMEN

Este artículo presenta la arquitectura basada en agentes inteligentes para la gestión de configuración de red, propuesta por los autores, en su intención por facilitar la compleja tarea de administración que debe realizar el personal responsable de la red, quien en ocasiones genera retrasos y costos adicionales debido a fallas humanas, poniendo en riesgo la productividad y competitividad organizacional. La arquitectura es validada por medio de un prototipo desarrollado en Java utilizando el framework JADE, que se prueba en un escenario real y permite demostrar que el uso de agentes inteligentes y ontologías facilitan el trabajo de gestión de configuración de la red.

PALABRAS CLAVE: Gestión de configuración de red, arquitectura, agentes inteligentes, ontología.

ABSTRACT

This paper presents a based intelligent agents architecture for Configuration Network Management, proposed by authors, in their intention to facilitate the complex management work, that to be performed by the staff responsible for the network, who sometimes generates delays and additional costs due to human failure, putting at risk organizational productivity and competitiveness. The architecture is validated through a prototype developed with JAVA using JADE framework, being tested in a real scenario, proving that the use of ontologies and intelligent agents facilitates the Configuration Network Management.

KEYWORDS: Configuration network management, architecture, intelligent agent, ontology.

1. INTRODUCCIÓN

En un contexto marcado por el crecimiento en tamaño y complejidad de las redes de computadoras, el concepto de gestión de red ha cobrado mayor importancia, al momento de administrar de manera eficaz, eficientemente y segura los recursos que la conforman con el propósito de asegurar su disponibilidad. Aunque existen múltiples soluciones en el mercado (Cisco 2012; Solarwinds, 2012), no se han encontrado herramientas que implementen agentes inteligentes para facilitar las tareas de administración de red, tales como: la gestión de la capacidad, aumento de la disponibilidad, supervisión y resolución de problemas, entre otras.

La gestión de configuración, como área funcional de la gestión de red incluye todas las tareas que tienen que ver con la recolección de información, modificación del comportamiento de los dispositivos y almacenamiento de la información que determina este comportamiento (Guerrero 2007; ITU-T, 2000). Actualmente, el crecimiento de las redes de computadoras y la heterogeneidad de los recursos y servicios de telecomunicaciones, conlleva a que actividades de mantenimiento de inventario de red, salvaguarda y restauración de la configuración ante la detección de errores, versiones de configuración, control de estado, entre otras, se tornen más complejas de ejecutar para el administrador de red, poniendo en riesgo la productividad y competitividad organizacional debido al error humano (Guerrero, 2007).

Es por eso que, con el objetivo de facilitar las labores de gestión, a comienzos de la década de los 90 aparecieron en escena los modelos de gestión de red integrada; como es el caso del protocolo simple de gestión de red SNMP (Simple Network Management) (Cisco 2009), el protocolo común de gestión de red CMIP (Common Management Information Protocol) (ITU-T, 1997), la interfaz de gestión de ordenadores de sobremesa o DMI (Desktop Management Interface) (Distributed Management Task Force, 2003) y la arquitectura común de agentes de peticiones de objetos o CORBA (Common Object Request Broker Architecture) (OMG, 2012), pero el uso de estos modelos trajo como consecuencia problemas de interoperabilidad debido a los diferentes fabricantes y arquitecturas, de ahí nace la gestión basada en Web de empresa o WBEM (Web Based Enterprise Management) que trata de establecer una arquitectura que permite la interoperabilidad entre modelos de gestión (Thomson, 1998).

Con el fin de aprovechar los avances alcanzados en la rama de la inteligencia artificial, en relación con el uso de las ontologías, para resolver problemas similares en el ámbito de la Web Semántica, se han aplicado estas técnicas en la gestión de redes como medio para compartir la semántica entre los modelos de gestión existentes (Guerrero, 2007; López, 2003), sin embargo, no abordan el área específica de la gestión red. También se propone el uso de agentes móviles para el monitoreo de redes LAN (Ríos y Sabogal, 2010), para facilitar la obtención de información de cada una de las estaciones de trabajo y así presentar dicha información en forma clara y fácil de comprender, de manera que pueda ser usada como soporte en la gestión de aspectos de la red (procesos, almacenamiento y memoria), no obstante, en esta propuesta no se utilizan ontologías, dificultando la implementación de los cambios requeridos para atender las nuevas exigencias de las tecnologías emergentes de las redes de nueva generación. Al respecto Sheng-Yuan Yang y Yi-Yen Chang (2010) proponen un nuevo sistema de gestión de red con ontología soportado en técnicas de multi-agentes, que aunque representa muchas ventajas, no explota el potencial de los agentes inteligentes ni centra su atención en una de las tareas específicas de la gestión de la red.

En este orden de ideas, queda en evidencia la necesidad de una arquitectura basada en agentes inteligentes para la gestión de configuración de red, que evite los inconvenientes generados por el error humano al poner en riesgo la productividad y competitividad organizacional, aprovechando al

máximo las ventajas que representa el uso de los agentes inteligentes de software y las ontologías como medio para la integración semántica de los distintos dominios de gestión existentes.

2. METODOLOGÍA

La arquitectura basada en agentes inteligentes para la gestión de configuración de red se construye a partir del proceso planteado por Bass (2003), que incluye las siguientes actividades: Delineación del proceso, diseño de la arquitectura, validación de la arquitectura y documentación de la arquitectura. Para cumplir con la primera actividad se define el modelo de dominio para el proceso de gestión de configuración de red. Dicho modelo se utiliza como punto de partida para definir la ontología según lo propuesto por Noy (2005): Definir el dominio y el alcance de la ontología, considerar la reutilización de ontologías existentes, enumerar términos importantes para la ontología, definir clases y jerarquía de clases, definir las propiedades de las clases (Slots), definir las facetas de los slots, y finalmente crear instancias.

El diseño de la arquitectura se elabora atendiendo el modelo propuesto por Kruchten (1995) y la validación de la arquitectura se hace por medio de un prototipo, que se construye aplicando la técnica de construcción de prototipos (Pressman, 2010), que incluye las siguientes actividades: se identifican los requerimientos, se construye y se revisa la maqueta, el cliente prueba la maqueta; estas actividades se repiten iterativamente para afinar el prototipo.

3. RESULTADOS

Los resultados obtenidos se presentan a partir del modelo ontológico y las vistas arquitectónicas: lógica, de escenarios y de despliegue, que posteriormente son validadas a través del prototipo utilizado en un escenario real de aplicación y cuyos resultados se comentan más adelante en el ítem de discusión.

3.1 MODELO ONTOLÓGICO

Una ontología es una descripción explícita y formal de conceptos en un dominio de discurso (clases o conceptos), propiedades de cada concepto describiendo varias características, atributos del concepto (slots, roles o propiedades) y sus restricciones (facetas o restricciones de rol). Una ontología junto con un conjunto de individuos de clases constituye una base de conocimiento (Noy y McGuinness, 2005). En este caso, el dominio lo constituye la gestión de configuración de red, los conceptos que describan el estado actual de la red, la gestión de inventario, el mantenimiento de directorios, la coordinación del esquema de nombres para nodos y aplicaciones y el control operacional de la red, como se observa en la Figura 1, donde se muestra cada uno de los conceptos con sus respectivos atributos y restricciones.

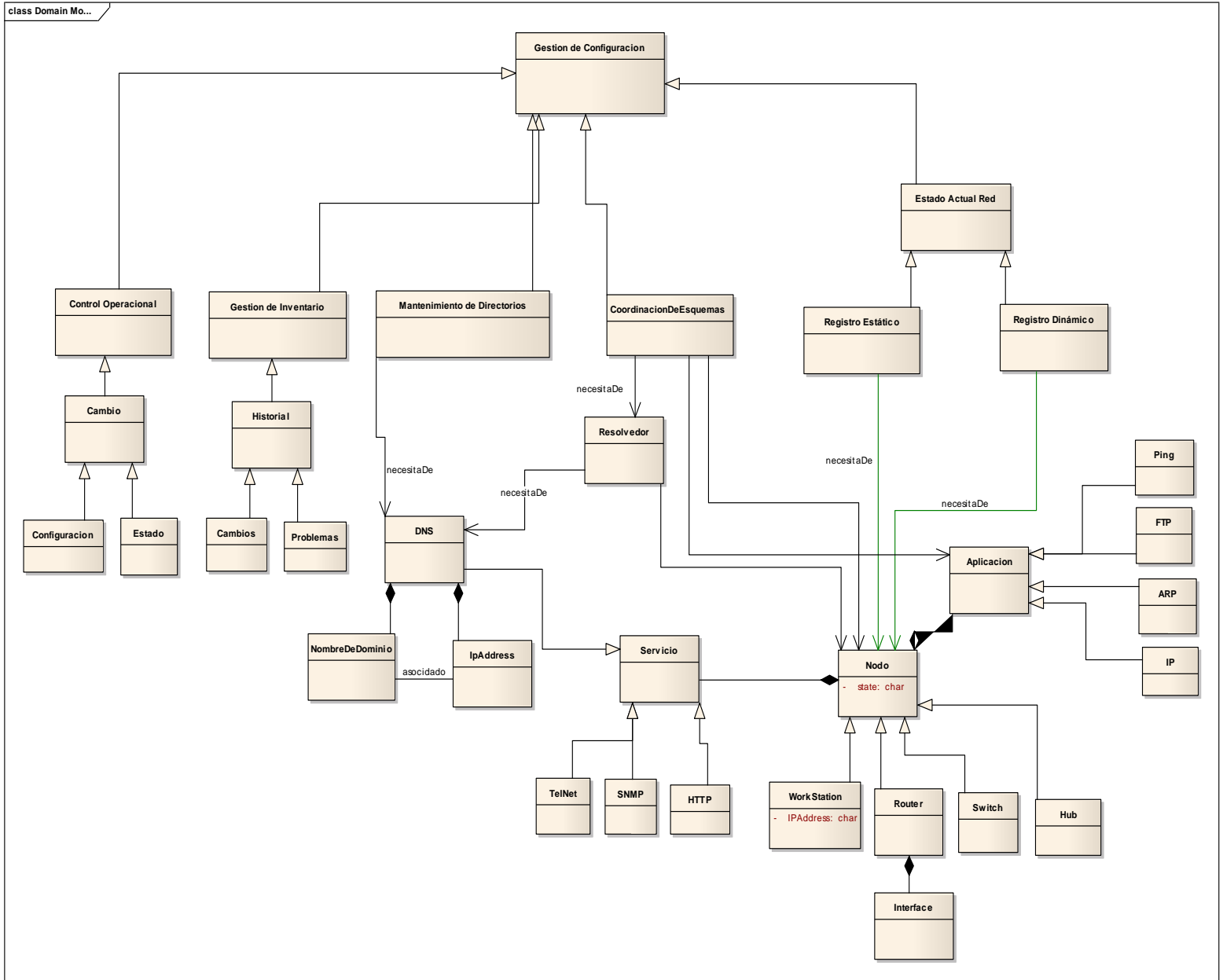


Figura 1. Modelo ontológico de gestión de configuración de red.

3.2 ARQUITECTURA

La arquitectura propuesta se representa utilizando el modelo de Vistas 4+1 (Kruchten, 1995), el cual describe la estructura y el comportamiento del sistema. Por medio de la vista de escenarios se dan a conocer las funcionalidades del sistema; mediante la vista lógica se indica cómo la estructura interna del sistema atiende dichas funcionalidades, finalmente se presenta la vista de despliegue con el fin de orientar la configuración y puesta en marcha del prototipo construido.

Vista de escenarios

El actor principal en el contexto de la configuración de red es el Administrador de la red, quien debe realizar actividades como: Solicitar registro estático o dinámica de la red; gestionar inventario para obtener la información correspondiente a los elementos que pertenecen a la red; consultar mantenimiento de directorios, con el fin de conocer los nombres de dominio de las estaciones de trabajo gestionadas, sus aplicaciones y la base de datos de nombres de dominios; coordinar esquemas de nombres, para conocer los esquemas de clasificación de los nombres de dominios en la red gestionada y controlar operaciones de red; cambiar la configuración de dispositivos para iniciar y detener elementos individuales. La vista de escenarios se presenta en la Figura 2.

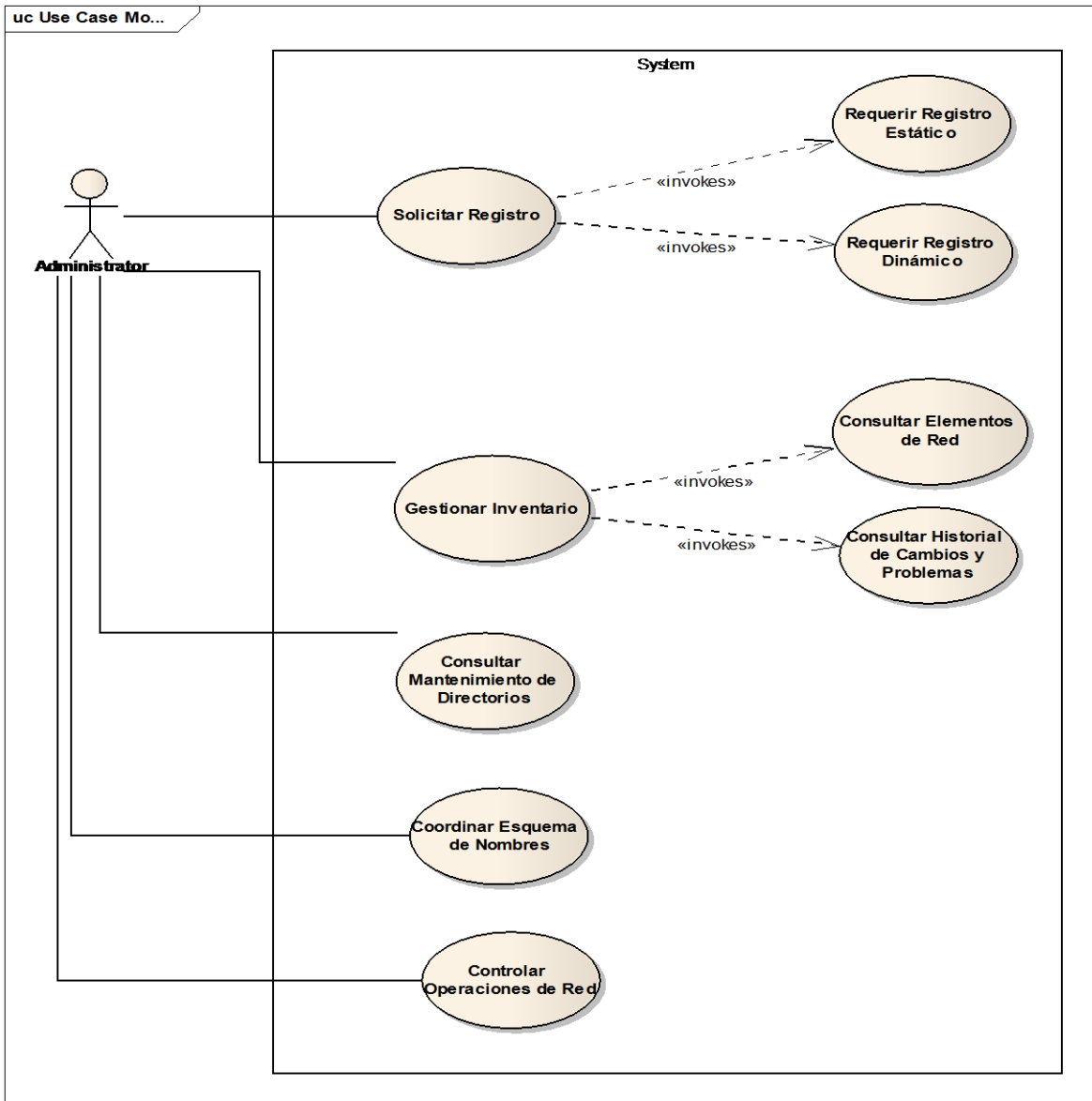


Figura 2. Vista de escenarios del sistema.

Vista lógica

El sistema está conformado por tres componentes que implementan su funcionalidad principal, que consiste en facilitar la gestión de configuración de la red. Los componentes son: Pizarra, mensajero y gestor. El componente pizarra es el encargado de solicitar un servicio, a su vez está conformado por tres subcomponentes: GUIAdministrador: contiene todas las clases y paquetes que permiten el desarrollo del entorno gráfico. AgentePizarra: se encarga de obtener las peticiones realizadas por el usuario e interactuar con el agente mensajero; IntérpreteDeResultados: se encarga de recorrer la ontología del sistema para descifrar los resultados.

El componente mensajero gestiona la petición y la lleva al componente gestor. Está conformado por el AgenteMensajero que se encarga de interactuar con el AgenteGestor a través del IntérpreteDePeticiones y el ReceptorDePeticiones, quien recibe las peticiones realizadas por el usuario y las suministra al AgenteMensajero. El componente gestor ofrece la interfaz gestionar red y está conformado por: El AgenteGestor: permite obtener la información solicitada por el usuario; IntérpreteDePeticiones: recibe las solicitudes que proporciona el AgenteMensajero y consulta la Ontología; el GestorDeResultados: obtiene y consulta la Ontología de Gestión para tabular la información consultada. La vista lógica del sistema se presenta en la Figura 3

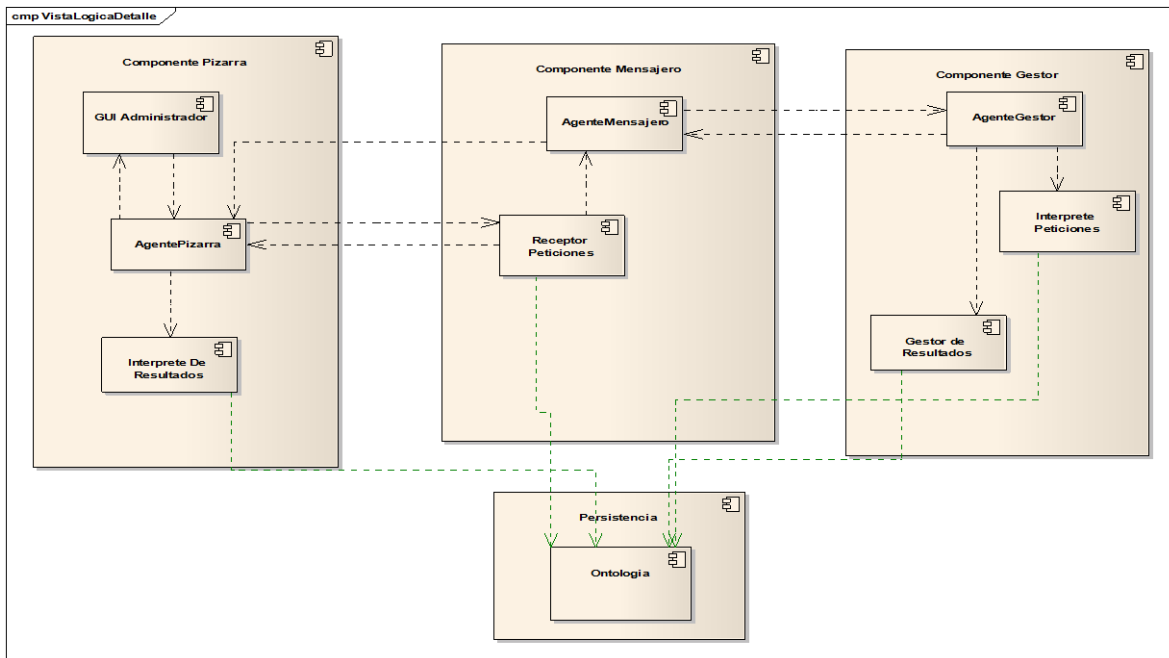


Figura 3. Vista lógica del sistema.

Vista de despliegue

Para el correcto funcionamiento del sistema se requieren por lo menos dos nodos conectado en red a través de un puente (Bridge), uno en calidad de servidor (Server) y otro que representa las estaciones de trabajo (WorkStation). El componente gestor es instalado en la estación de trabajo que debe contar con la máquina virtual de java, mientras que los componentes pizarra y mensajero se instalan en el servidor quien debe además debe contar con el Framework Jade (java Agentdevelopment) y la ontología, como se observa en la Figura 4.

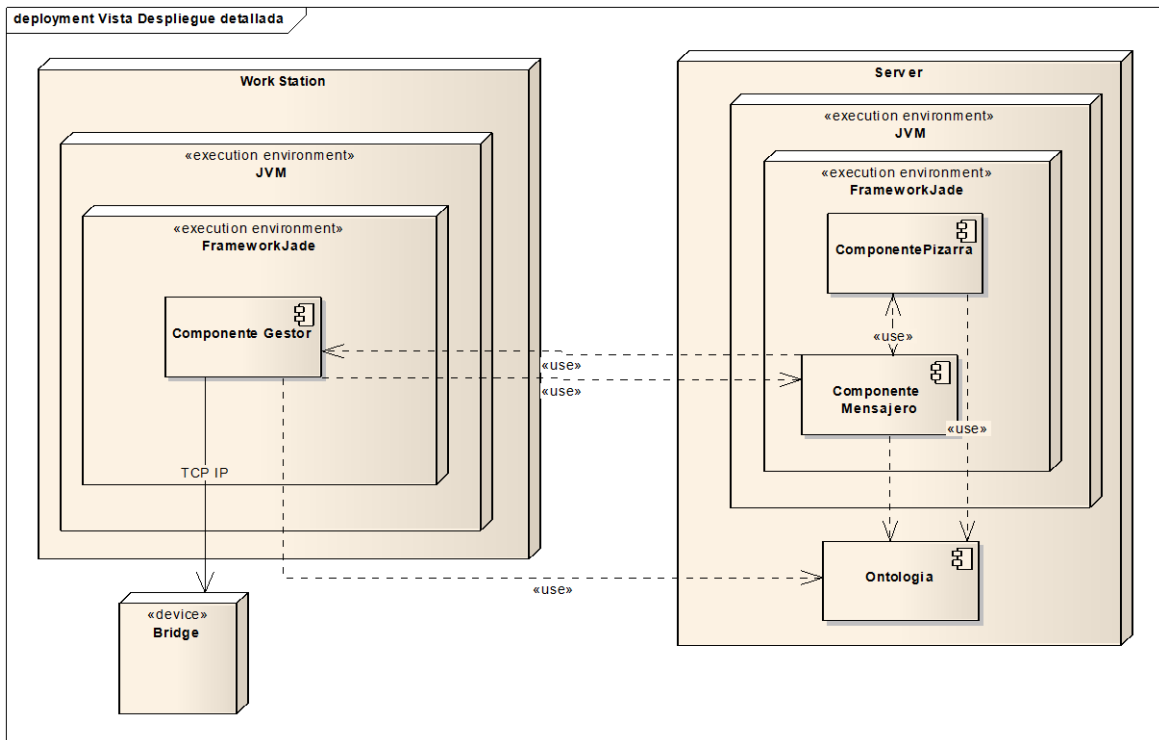


Figura 4. Vista de despliegue del sistema.

4. DISCUSIÓN DE RESULTADOS

La validación de la arquitectura se realizó a través de la construcción y aplicación de un prototipo de agente inteligente que soporta las funcionalidades que comprende la gestión de configuración de red. Para el caso práctico, se definió como escenario de ejecución los laboratorios de software y de redes del programa de Ingeniería de Sistemas de la Universidad de Cartagena, utilizados por la comunidad académica para el desarrollo de sus actividades y actualmente carecen de un control a nivel de gestión de configuración de red, generando inconvenientes de seguridad y permitiendo que los estudiantes utilicen dicho recurso para actividades ajenas al proceso de formación, afectando negativamente el proceso de aprendizaje.

En los laboratorios hay 60 computadores conectados en red, distribuidos en tres salas de cómputo y un salón para el semillero de estudiantes, bajo la administración de un monitor. En este contexto se definió en calidad de servidor el equipo utilizado por el monitor de la sala, con el fin de permitirle el control de configuración de la red, además se utilizaron 25 ordenadores en calidad de nodos de la red, en cada uno de los cuales se instaló el componente gestor, luego se procedió a realizar el proceso de envío/respuesta de peticiones desde el componente pizarra. Se realizaron tres peticiones de registro estático y dos de registro dinámico con los resultados que se relacionan en la Tabla N° 1.

Tabla 1. Registro de peticiones realizadas.

| Iteración | Tiempo de respuesta (seg.) | Nodos que contestaron | Petición de registro |
|-----------|----------------------------|-----------------------|----------------------|
| 1 | 50 | 14 | Estático |

| | | | |
|---|----|----|----------|
| 2 | 40 | 7 | Dinámico |
| 3 | 45 | 20 | Estático |
| 4 | 35 | 25 | Dinámico |
| 5 | 44 | 16 | Estático |

Al observar los resultados obtenidos, se hace evidente que la cuarta iteración corresponde al mejor escenario porque obtuvo el menor tiempo de respuesta para la totalidad de nodos, en contraste con la segunda iteración, donde sólo se obtuvo respuesta de una cuarta parte de los nodos utilizando un mayor tiempo. Sin embargo, para el caso de estudio se logró el objetivo, porque el prototipo desarrollado a partir de la arquitectura propuesta demostró que permite la captura, tanto del registro estático (Ver Figura 5) como del registro dinámico, lo que posibilita la gestión automática de configuración de la red.

En este orden de ideas, la solución planteada implica ventajas para el administrador de la red. La sistematización del proceso de recolección de información en la red dejando atrás la práctica manual, la optimización de los tiempos de respuesta orientados a disminuir los lapsos de espera y la ejecución de tareas de gestión de configuración de forma independiente a los mecanismos de gestión de los recursos; conforman los aspectos más relevantes, todas ellas orientadas a facilitar la labor de gestión de configuración de red ejercida por el administrador, centrando la atención en mejorar el acceso a la información y con ello la obtención de conocimiento útil, con lo cual se torna el proceso de gestión más eficiente.

The screenshot shows the 'GESCONFIG.RED' application interface. A central window titled 'R. ESTÁTICO' displays a table with the following data:

| IP address... | Host | Mac addr... | Net Mask... | Gateway... | #Interface... | Descripci... | Version S... | Arquitectu... |
|---------------|-------|-------------|-------------|--------------|---------------|--------------|--------------|---------------|
| 10.10.7.30 | SIS30 | 00:0F:FE... | 255.255... | 10.10.6.1... | 2 | Microsoft... | 5.1 | x86 |
| 10.10.7.17 | SIS17 | 00:0F:FE... | 255.255... | 10.10.6.1... | 2 | Microsoft... | 5.1 | x86 |
| 10.10.7.21 | SIS21 | 00:0F:FE... | 255.255... | 10.10.6.1... | 2 | Microsoft... | 5.1 | x86 |
| 10.10.7.20 | SIS20 | 00:0F:FE... | 255.255... | 10.10.6.1... | 2 | Microsoft... | 5.1 | x86 |
| 10.10.7.19 | SIS19 | 00:0F:FE... | 255.255... | 10.10.6.1... | 2 | Microsoft... | 5.1 | x86 |
| 10.10.7.8 | SIS08 | 00:0F:FE... | 255.255... | 10.10.6.1... | 2 | Microsoft... | 5.1 | x86 |
| 10.10.6.11 | SIS26 | 00:0F:FE... | 255.255... | 10.10.6.1... | 2 | Microsoft... | 5.1 | x86 |
| 10.10.7.16 | SIS16 | 00:0F:FE... | 255.255... | 10.10.6.1... | 2 | Microsoft... | 5.1 | x86 |
| 10.10.7.29 | SIS29 | 00:0F:FE... | 255.255... | 10.10.6.1... | 2 | Microsoft... | 5.1 | x86 |
| 10.10.7.13 | SIS13 | 00:0F:FE... | 255.255... | 10.10.6.1... | 2 | Microsoft... | 5.1 | x86 |
| 10.10.7.11 | SIS11 | 00:0F:FE... | 255.255... | 10.10.6.1... | 2 | Microsoft... | 5.1 | x86 |
| 10.10.7.6 | SIS06 | 00:0F:FE... | 255.255... | 10.10.6.1... | 2 | Microsoft... | 5.1 | x86 |
| 10.10.7.15 | SIS15 | 00:0F:FE... | 255.255... | 10.10.6.1... | 2 | Microsoft... | 5.1 | x86 |
| 10.10.7.5 | SIS05 | 00:0F:FE... | 255.255... | 10.10.6.1... | 2 | Microsoft... | 5.1 | x86 |
| 10.10.7.30 | SIS30 | 00:0F:FE... | 255.255... | 10.10.6.1... | 2 | Microsoft... | 5.1 | x86 |
| 10.10.7.28 | SIS28 | 00:0F:FE... | 255.255... | 10.10.6.1... | 2 | Microsoft... | 5.1 | x86 |

The interface also includes a 'REGISTROESTATICO' label at the bottom left and a 'UDC' logo at the bottom right.

Figura 5. Información de registro estático

5. CONSIDERACIONES Y CONCLUSIONES

Se definió una arquitectura basada en agentes inteligentes y ontologías para la gestión de configuración de red, conformada por tres componentes que implementan su funcionalidad principal: Pizarra, mensajero y gestor. Dicha arquitectura se validó a través de un prototipo desarrollado en Java, utilizando como Framework JADE para la construcción de los agentes, aprovechando su facilidad de instalación y uso, eficiencia y eficacia en el manejo de los recursos informáticos y la posibilidad de que el prototipo pueda ser extendido a dispositivos móviles.

Al realizar las pruebas con el prototipo desarrollado se comprobó que los agentes son funcionales y cumplen con los parámetros establecidos en la arquitectura y que la ontología usada es funcional, brindando entre otros beneficios: Eficiencia en el manejo de los recursos informáticos debido al uso de la tecnología de agentes y ontologías, y flexibilidad ante cambios que se puedan presentar en las labores de gestión de configuración de red ya que esta tarea se puede hacer mediante la actualización de la ontología.

Por lo tanto, se demostró que la arquitectura basada en agentes inteligentes para la gestión de configuración de red, propuesta por los autores, hace más fácil la compleja tarea de administración de la red, evitando retrasos y costos adicionales generados por fallas humanas, y haciendo posible un incremento de la productividad y competitividad organizacional.

El grupo de investigación continuará trabajando sobre la arquitectura para identificar las causas del comportamiento irregular de las peticiones de gestión que se hicieron en las pruebas y mejorar los tiempos de respuesta, a fin de minimizar el impacto en el desempeño de la aplicación. Igualmente se extenderá el desarrollo para que el sistema pueda ser utilizado en dispositivos móviles.

REFERENCIAS

Bass, Len. Clements, Paul, Kazman, Rick. (2003) Software architecture in practice. Second edition. Addison Wesley.

Cisco. Cisco application networking manager. Disponible en:
<http://www.cisco.com/en/US/products/ps6904/index.html>. visitado febrero 12 de 2012

Cisco Systems, I. (17 de 12 de 2009). Wiki Cisco. Recuperado el 31 de 7 de 2012, de Wiki Cisco:
http://docwiki.cisco.com/wiki/Simple_Network_Management_Protocol

Distributed Management Task Force, I. (10 de 01 de 2003). DMTF. Recuperado el 31 de 07 de 2012, de DMTF: <http://dmf.org/sites/default/files/standards/documents/DSP0005.pdf>

Guerrero Casteleiro, A. (2007). Biblioteca Universitaria Politécnica, Archivo Digital UPM. Recuperado el 1 de agosto de 2010, de Especificación del comportamiento de gestión de red mediante ontologías: http://oa.upm.es/909/1/ANTONIO_GUERRERO_CASTELEIRO.pdf

International Telecommunications Union - Telecommunications standardization sector (ITU-T), I. t. (10 de 1997). Open Systems Interconnection - Common Management Information Protocol: Specification. Recomendación X.710.

International Telecommunications Union - Telecommunications standardization sector (ITU-T), I. T.-T. (02 de 2000). Overview of TMN Recommendations. Recomendación M.3000.

Kruchten, P. (1995) "The 4+1 view model of architecture," IEEE Software 12(6),

López de Vergara Méndez, J. E. (2003). Scientific literature digital library and search engine. Recuperado el 5 de agosto de 2010, de especificación de modelos de información de gestión de red integrada mediante el uso de ontologías y técnicas de representación del conocimiento: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.77.8694&rep=rep1&type=pdf>

Noy, N., & McGuinness, D. (19 de septiembre de 2005). Desarrollo de ontologías 101: Guía para crear tu primera ontología. Recuperado el 2 de septiembre de 2010, de Desarrollo de Ontologías 101: Guía para crear tu primera ontología: http://protege.stanford.edu/publications/ontology_development/ontology101-es.pdf.

Object management group, I. (26 de 07 de 2012). Object management group. Recuperado el 31 de 07 de 2012, de Object management group: <http://www.omg.org/gettingstarted/corbafaq.htm>

Pressman, R. (2010). Ingeniería de software: Un enfoque práctico. En R. Pressman, Ingeniería de Software: Un enfoque práctico (pág. 959). Madrid: Mc Graw Hill.

Ríos, L. E., & Sabogal, M. G. (2010). Sistema de agentes móviles para el monitoreo en la gestión de redes. Bogotá: Universidad Distrital Francisco José de Caldas.

Object management group, I. (26 de 07 de 2012). Object management group. Recuperado el 31 de 07 de 2012, de Object management group: <http://www.omg.org/gettingstarted/corbafaq.htm>

Shang, S.-Y., & Chang, Y.-Y. (09 de 2010). A new network management system with ontology-supported multi-agent techniques. IEEE Computer Society.

Solarwinds. Solarwinds It management & performance monitoring software- Visitado marzo 3 de 2012. disponible en: <http://www.solarwinds.com>.

Thompson, J. (1998). Web-based enterprise management architecture. IEEE Communications Magazine, 80-86