
Una propuesta de sistema inteligente para apoyar en el diagnóstico de la tbC utilizando ontologías y agentes de software

Luzmila Pró Concepción, Augusto Cortez Vásquez, Nora La Serna Palomino,
Ulises Román Concha, Lázaro Mota Alva

Universidad Nacional Mayor de San Marcos

Facultad de Ingeniería de Sistemas e Informática

lproc2003@hotmail.com, cortez_augusto@yahoo.fr, nlasernap@unmsm.edu.pe,
nromanc@yahoo.es, lmotaa@unmsm.edu.pe

ReSUMeN

El presente estudio plantea “Una propuesta de sistema inteligente para apoyar en el diagnóstico de la TBC utilizando ontologías y agentes de software” Fue realizado en varias etapas: el estudio teórico de los “Fundamentos de Ingeniería de la Web: Ontologías, Web Semántica y Agentes de Software”, el estudio de la enfermedad caso de la TBC, los síntomas, estadísticas en el Perú y en el mundo, datos obtenidos del Ministerio de Salud del Perú y de la Organización Mundial de la Salud, el análisis y el diseño del proyecto a fin de desarrollar un prototipo que ayude en el diagnóstico de la TBC, mediante ontologías, web semántica y agentes de software, para finalizar se han presentado las conclusiones.

Palabras Claves: sistemas inteligentes, ontologías, web semántica, agentes de software, diagnóstico de la TBC

AbStRAcT

The present study of investigation: “A proposal of Intelligent Systems to support in the TBC Diagnostic intervening Ontology's and Software Agents”, was realized in several stages: the theory study of “The Fundaments of Web Engineering: Ontology, Semantic Web, and Software Agents”, the study of the sickness, case of the TBC, the symptoms and statistics cases in Perú and in the world, the get data of Health Ministry of Perú and the World Health Organization, and the analysis and design of the project for development a prototype to support in the diagnostic of the TBC, intervening Ontology's, Semantic Web and Software Agents, for to finish was presented the conclusions.

Keywords: intelligent systems, ontology's and software agents, diagnostic of the TBC

1. INTRODUCCIÓN

El presente proyecto se realizó en varias etapas: primero se realizó el estudio teórico que fue presentado como un artículo intitulado: "Fundamentos de Ingeniería de la Web: Ontologías, Web Semántica y Agentes de Software" [12], se tiene como antecedentes al proyecto: "Análisis de historia clínica para la Ictericia Obstructiva Extrahepática", financiado por el FINCYT [9] ambos estudios se basan en la Teoría de Agentes, luego se estudió a la TBC, en este caso se aplicando las buenas prácticas del modelamiento de agentes AUML, así como también un estándar de los datos que se le denomina ontología.

El análisis y el diseño del estudio de un prototipo con sistema multiagentes para la gestión de citas para apoyar en el diagnóstico de la TBC a través de Informes de Ministerio de Salud (MINSA). Se analizó también la situación actual del servicio de emergencia de los Hospitales Dos de Mayo y Arzobispo Loayza en cuanto a la gestión citas en particular para el TBC.

Para ello nos ponemos a realizar lo siguiente:

- Analizar el servicio de emergencia en los hospitales Dos de Mayo y Arzobispo Loayza y en particular la gestión de los servicios en el área de la TBC.
- Modelar la arquitectura del sistema multiagentes para la gestión de servicios usando la Metodología Ingenias.
- Diseñar un prototipo del sistema multiagentes para la gestión de citas. Diseñar un Modelo de apoyo en el diagnóstico de la TBC.

2. MARCO teórico

El presente estudio presentó en un artículo los fundamentos teóricos "Fundamentos de Ingeniería de la Web: Ontologías, Web Semántica y Agentes de Software" [12], sin embargo en este artículo presentamos sobre los agentes de software y la definición de herramientas que utilizará el proyecto para desarrollar el prototipo.

2.1. Agentes de Software

Son un conjunto de software que pueden automatizar los procesos y que pueden ser usados dinámicamente para describir, descubrir, estructurar y componer servicios. El uso de los agentes de Soft-

ware y sistemas multiagentes permite el desarrollo de una nueva generación de aplicaciones, basadas en tecnologías emergentes y en un conjunto de estándares abiertos para la web, en este contexto, la demanda por aplicaciones basadas en agentes de software se ha incrementado con aplicaciones de software en áreas como la salud. Un agente está situado en un entorno, que sería la propia red (Internet). La definición de Russell, (1996) considera a un agente como una entidad que percibe y actúa sobre un entorno. Los agentes no son desarrollados de forma independiente sino como entidades que constituyen un sistema denominado **multi-agente**. Los agentes pueden interactuar e informar, consultar a otros agentes, «hablar» entre ellos, la comunicación entre agentes se realiza mediante (ACL—*Agent Communication Language*). Una entidad software es un agente si tiene tres propiedades: Autonomía: capaz de elegir sus semejantes, controlar, asegurar o recusar una petición; Interacción: comunicarse en su ambiente con otros agentes por medio de sensores; Adaptación: adaptarse, modificar su estado, su comportamiento de acuerdo con nuevas condiciones. FIPA (*Foundation for Intelligent Physical Agents*) define un agente como "Un proceso computacional que implementa una funcionalidad comunicativa autónoma en una aplicación" (FIPA, 1997). Según ARPA, y su proyecto KSE (*Knowledge Sharing Effort*), OMG (*Object Management Group*) con su proyecto MASIF (*Mobile Agent System Interoperabilities Facility*) y Agent Society han definido estándares para la construcción de agentes (arquitecturas, protocolos de comunicación, aplicaciones, etc.) pero la definición dada por FIPA tiene mayor aceptación.

2.2. Herramientas

Las herramientas que se utilizan en el demo son: Eclipse IDE for Java Developers, Base de datos Mysql 5, Plataforma JADE, INGENIAS, DIA (Diseño de AUML), JSP, Servlet, hojas de estilos (CSS), sistema operativo Linux UBUNTU.

2.2.1. eclipse Ide for Java developers

Es una herramienta creada para Java EE y Aplicaciones Web, incluye Java IDE herramientas para Java EE, JPA, JSF Mylyn y otras. Eclipse IDE for Java Script Web Development es una herramienta para Java Script IDE, y HTML, CSS y XML (15).

2.2.2. base de datos MySQL 5

Es un sistema de Gestión de Base de Datos Relacional incluye bugs reportados y seleccionados y en lista cómo MySQL 5. El Software MySQL proporciona un servidor de Base de Datos SQL (Structured Query Language) muy rápido multihilo, multiusuario y robusto. El servidor MySQL está diseñado para entornos críticos y distribuidos [11].

2.2.3. Plataforma JADE – Java Agent development environment

Está compuesta de la Plataforma FIPA desarrollado por FIPA (*Foundation for Intelligent Physical Agents*) para la ejecución de agentes: Un conjunto de paquetes para la programación de agentes FIPA, es 100% Java con el Java Development Kit JDK 1.4. Creación básica de agentes, Programación de comportamientos de agentes en behaviours, FIPA-ACL (*Agent Communications Language*) para envío y recepción de mensajes, Clases para programar protocolos FIPA (y no FIPA), códigos (FIPA-SL (*Semantic Language*)) para la gramática de la comunicación de agentes, FIPA-RDF (*Resource Description Framework*) ve la comunicación y secuencia de mensajes entre agentes, etc.). La Plataforma FIPS (*AMS Agent Management Specification*), facilitador de directorio y MTS (*Message Transport Service*). La plataforma JADE puede ejecutarse en una o más hosts distribuidas administradas por JVM (*Java Virtual Machine*), en forma concurrente e intercambiar mensajes, Organizar en contenedores en uno principal: AMS (*Agent Management Specifications*) y en la DF (*clase Jade Domain*) y el registro *mi* están localizados ahí, *n* containers. [7]

Servicios Básicos de Directorio: El Directorio al igual que FIPA es un servicio accesible a través de JADE, Domain DF Services, accede a las páginas amarillas desde una interface para los servicios de tipo: register, dregister, modify y search.

La clase agentes: Programan en agentes JADE, definen una clase Java, representa al agente, determina y codifica los comportamientos que va a manifestar, hace que herede de la clase JADE, Core Agent, Programa sus métodos Setup().takeDown, para ejecutar el agente se hace desde GUI de JADE u otro programa. JADE propone un modelo de agente, single threaded y añade un nivel de scheduling sobre la única thread a nivel de comportamientos.

Scheduling de comportamientos: Cada agente tiene su cola de comportamientos activos, el cuerpo de acciones de un comportamiento se programa redefiniendo el método action(), cuando el método anterior finaliza dependiendo del tipo de comportamiento, el scheduling le saca de la cola, lo vuelve colocar al final, un comportamiento puede bloquearse mediante block() hasta que lleguen más mensajes al agente, el bloqueo significa que cuando action() termina se coloca en una cola bloqueada, si llega un nuevo mensaje sale de esa cola y se coloca al final de la cola de comportamientos activos. La clase JADE.long.ACLMessage es la base para la composición de mensajes (Met, set y get), son todos los parámetros de mensaje FIPA. Los métodos agentes send (...), agent receive (...) y el agentblockingReceive() es para envío y recepción de mensajes. La clase JADE.long.ACLMessageTemplate realiza el matching de mensajes. FIPA define protocolos de interacción, son manejadores basados en clases Jade,proto. AchieverREInitiator, Jade.proto.AchieverRESPonder necesarios para implementar FIPA-Request, FIPA-Query, FIPA-Request-When, FIPA-securing y FIPA Trucking-suscribe, FIPA-contract-net dispone de jade.protoContractNetInitiator y jade.protoContract.NetResponder [16].

Ontologías Básicas en JADE: JADE permite el manejo de ontologías para representar el dominio de una aplicación mediante conceptos, predicados, acciones y agentes, etc. Intercambiar elementos de la ontología entre agentes (en el cuerpo del Mensaje FIPA) usa un sistema gestor de ontologías Protégé 2000 para el modelado y generación de código Java JADE. El Manejo de ontologías desde Protégé, consiste en seguir una secuencia de pasos: 1. Conceptualizar el problema: definir elementos que participarán en la ontología, definición de agentes, definición de Interacciones que tendrán lugar entre los agentes; 2. Diseño de la ontología con Protégé; 3. Programación de protocolos de interacción con JADE en forma de comportamientos e integrando el código generado por Protégé; y 4. Programación de Agentes [1].

2.2.4. eNIAS

INGENIAS Development Kit (IDK), facilita el desarrollo de MAS, soporte de INGENIAS Development Process (IDP), ayuda a los desarrolladores, usa el concepto de agentes en el contexto de un sistema de desarrollo y

de prácticas de INGENIAS, como un resultado este proceso propone un conjunto de actividades que puede ser organizado. El IDP es un conjunto de herramientas y técnicas de implementación. El Metamodelo IDK INGENIAS define y determina que es un agente y que es un MAS. Este modelo es público y es accedido por INGENIAS METAMODEL v1.0, (4)

2.2.5. Diseño de Lenguaje De Modelamiento Unificado para Agentes (AUML)

El paradigma de agentes se basa en algoritmos, técnicas comerciales como frames y el paradigma orientado a objetos. Los agentes inteligentes es un componente de software o hardware, capaz de adecuar una forma autónoma para completar tareas de parte del usuario. Los agentes se agrupan en categorías: De colaboración, de interfaz, móviles, de información, o Internet, reactivos, híbridos, inteligentes. Atributos de los agentes son: Autonomía: operan sin intervención humana, Movilidad: se mueven en diferentes ambientes, Inteligentes: reconocen eventos, determinan su significado y a partir de ahí llevan a cabo sus acciones, BDI (Believe, Desire, Intentions), creencia: es conocimiento, deseos: son objetivos que tiene que cumplir, intenciones: son planes que llevará a cabo, Reactividad: perciben su entorno, Proactividad: responden a un estímulo, Sociabilidad: comunicación con otros agentes o puede ser también con un ser humano.(2)

Metodología Prometheus: Es una metodología de desarrollo de agentes inteligentes creado por Padgham y Winkoff, define un lenguaje de modelado, son parte de los fundamentos de AUML. Prometheus se ha desarrollado en forma conjunta con Agent Oriented Software, su ambiente de Desarrollo PDT (Prometheus Design Tool). La arquitectura Prometheus se aplica en 3 etapas: Especificación del sistema, Arquitectura del sistema, Diseño del sistema.

2.2.6. JSP, Servlet

JSP Servlet Java, son la respuesta de la tecnología a la programación CGI (*Common Gateway Interface*) son programas que se ejecutan en un servidor Web. La página Web se basa en datos enviados por los usuarios. Ejemplo páginas de resultados de motores de búsqueda y de programas que procesan pedidos desde los sitios de comercio electrónico. Java Server Pages (JSP) es una tecnología, permite mezclar HTML estático con HTML dinámico. (14)

2.2.7. Sistema operativo Linux Ubuntu

Linux es un sistema operativo compatible con UNIX, Linux es un software libre. El sistema operativo Linux lo forma un núcleo del sistema (Kernel), posee un gran número de programas y librerías que hacen posible su utilización, Linux es distribuido por el proyecto GNU. Linux es un sistema operativo completo multitarea y multiusuario al igual que UNIX. Linux implementa todo lo necesario para trabajar como TCP/IP por el puerto de serie. LINUX UBUNTU, se ha convertido en la distribución de Linux mas conocida, se usa con facilidad y seguridad, cuenta con su propio gestor de paquetes para las instrucciones de un nuevo programa. (8)

3. Presentación del caso de estudio

3.1. Caso de estudio: La tuberculosis

La tuberculosis (TBC) es una enfermedad infecciosa que suele afectar a los pulmones y es causada por la bacteria *Mycobacterium tuberculosis*, y puede afectar a otras partes del cuerpo. Por ejemplo: los ganglios linfáticos, los riñones, los huesos, las articulaciones, etc. (TBC extra pulmonar). Esta enfermedad puede afectar a personas de cualquier edad, las personas con mayor riesgo son aquellas cuyos sistemas inmunológicos están débiles. Los síntomas de la TBC son: fiebre leve, sudoración nocturna, fatiga, pérdida de peso y tos persistente. Si la TBC se presenta suele manifestarse 2 o 3 meses después de la infección o años después. Una persona con TBC puede contagiar la enfermedad mientras no sea sometida a un tratamiento apropiado. El plan de medicación debe ser determinado por un médico. Los programas de tratamiento (DOT) para los pacientes con TBC. Para evitar el contagio de la TBC, los pacientes con TBC se deben cubrir la boca y la nariz al toser y que tomen todos los medicamentos para la TBC, según las indicaciones del médico.

3.1. La tuberculosis en el Perú

El control de la tuberculosis en el Perú, data del año 1940, el Programa Nacional de Control de la Tuberculosis basado en la vacunación con BCG (Gironde, 1991). En 1960 el tratamiento utilizó Isoniacida (H) y Estreptomina (S) otorgada a los pacientes de manera gratuita. A fines de 1970, el Ministerio de Salud aplicó esquemas para ser autoadministrado en casi en toda su red con Rifampicina (R), Isoniacida (H), Pirazinamida (Z). Entre 1980 y 1985 no hubo una adecuada cobertura

Años	Nº de enfermos con tbc	Nº de enfermos con tratamiento	% de Cobertura
1980	16,011	7,000	43.7
1981	21,925	5,000	22.8
1982	21,579	6,000	27.8
1983	22,753	No hay datos	-----
1984	22,792	8,000	35.0
1985	24,438	13,000	53.0
Total	129, 438	39,000	30.1

Fuente Quimper W. Tuberculosis en el Perú situación actual y factores de riesgo TM 1988 TM UPCH.

tabla 1. Número de enfermos con TBC que recibieron tratamiento en el Perú entre 1980 a 1985.

Resultados	Número de pacientes tratados	% de enfermos tratados
Abandono el tratamiento	1027	41
	144	6
Fallecidos	111	4
Recaída	57	2
Presuntamente curados	1171	47

Fuente Philip C. Hopewell – Hernández Robert and Bernardus Ganter Operational evaluation for tuberculosis Results for <<standards>> 12 Month regimen in Perú Am. Rev Respir. dis 1984, 129; 439-443.

tabla 2. Evaluación de la Eficiencia del Tratamiento Antituberculosis en el Perú en 1988 (Esquema 2SD/10H2S2 supervisado o 2SD /10D autoadministrado).

para detectar los casos y dar tratamiento gratuito a los pacientes. Ver Tabla 1.

Se realizó una Evaluación de la Eficiencia del Tratamiento Antituberculosis en el Perú en 1988 (Esquema 2SD/10H2S2 supervisado o 2S/10D autoadministrado.). Ver Tabla 2.

En 1990 sólo el 25% de los servicios del Ministerio de Salud desarrollaron acciones para el diagnóstico y tratamiento de la TBC, en la actualidad el 100% de los mismos garantizan el acceso al diagnóstico y al tratamiento gratuito La situación de la TBC en el país presenta una tendencia decreciente, no es uniforme a lo largo del territorio, se concentra en Lima, Callao, Ica, Tacna, Madre de Dios, Loreto y Ucayalí. El Perú en control de la TBC, tiene una amplia experiencia, El MINSA ha liderado el proceso, de control de la TBC. El

estado tiene un compromiso político (incremento de la partida presupuestaria) para el sector salud destinado a luchar contra la TBC desde el 2006, las comisiones elaboran normas para las buenas prácticas de atención a las personas con TBC, y la apertura de aportes técnicos y financieros de alianzas estratégicas, la sociedad civil ha logrado que el Control de la TBC sea prioritario y se incluya en la agenda publica del estado (a nivel gobierno, regiones, localidades, hospitales y centros de salud). Tabla 3.

En los últimos 20 años en el sector salud han ocurrido cambios, convocando a profesionales con experiencia, a fin de mejorar los conocimientos y destrezas del personal hacia una cultura de calidad de atención. [8]. Según datos de la Organización Mundial de la Salud (OMS), el Perú ocupa el 7º lugar del mundo en TBC.[17].

Periodos	Logro de Control de la tbc
De 1980 (Primer semestre) a 1990	Década perdida
De 1990 a 1995	Etapas de Normalización y estructuración de Equipos Técnicos por niveles
De 1996 al 2000	Etapas de Implementación y Consolidación
De 2001 al 2010	Etapas de sostenibilidad y ampliación

Fuente TBC en el Perú New Paradigms facing in New Milenium 1999.

tabla 3. Etapas por las que ha atravesado el Control de TBC en el Perú

4. Modelo de diseño del Proyecto

4.1. Análisis y diseño de una gestión de citas en un hospital

El presente demo: módulo de gestión de citas, considerando desde que el paciente llega al hospital y suponiendo tenga TBC. El paciente entra a la gestión de

citas, verifica: la hora de su cita y el médico que atenderá su cita. Es un análisis básico de sistemas que utiliza AUML, Ingenierías en el Diseño (Diagrama 1).

Se presentan Diagrama de Clases, de Colaboración, Diagrama de Tareas. En el siguiente se diseña la BD con la ayuda del Diagrama de Clases, (Diagrama 4).

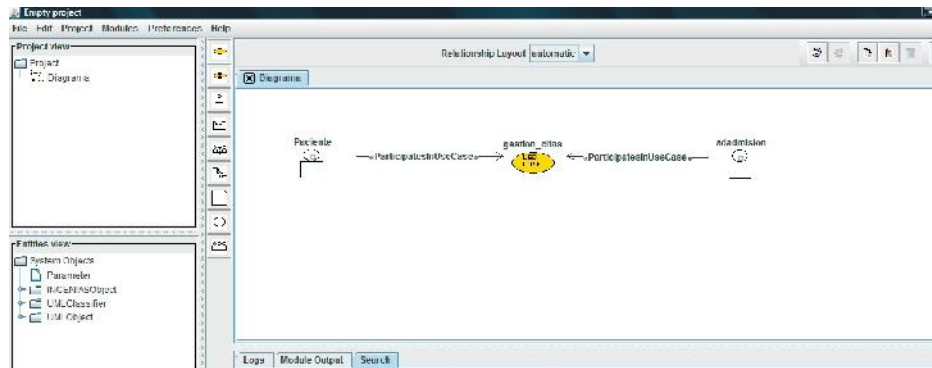


Diagrama 1. Diagrama Ingenierías AUML (Caso de Uso)

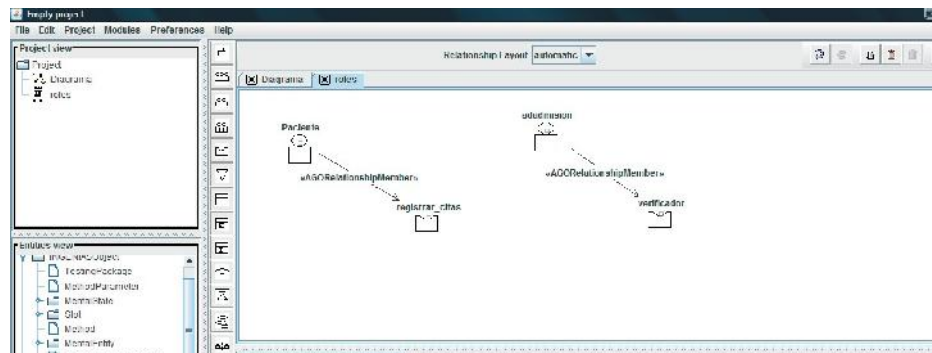


Diagrama 2. Diagrama de Roles.

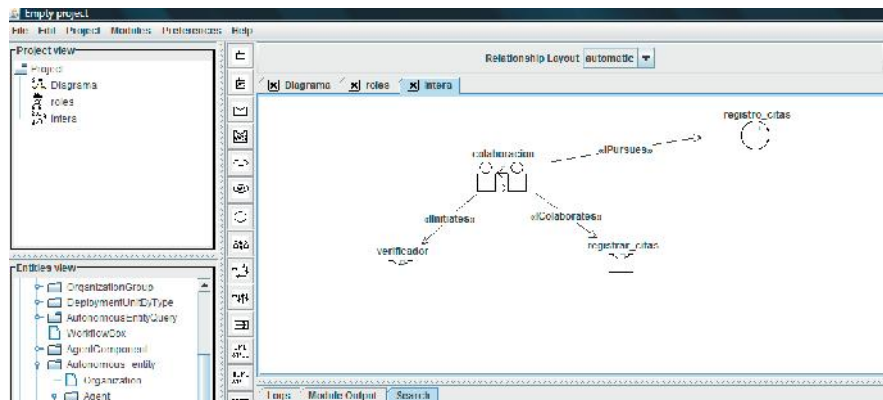


Diagrama 3. Diagrama de Interacción.



Diagrama 4. Diagrama de Clases.

En el desarrollo de la Base Datos Relacional, el agente puede insertar y extraer data, se desarrollan seis ta-

blas: de categorías, citas, históricos, médico, pacientes y usuario.

En el siguiente se muestran los procesos almacenados que serán aplicados (Diagrama 5).

Para diseñar se usa AXML de agentes: herramienta INGENIAS Development Kit (IDK). (4)

4.2. Modelo del Prototipo

En el modelo del prototipo se usan seis plantillas, la introducción, con los objetivos de la investigación, registro del paciente, la Figura 1 se muestra la asignación de cita al paciente.

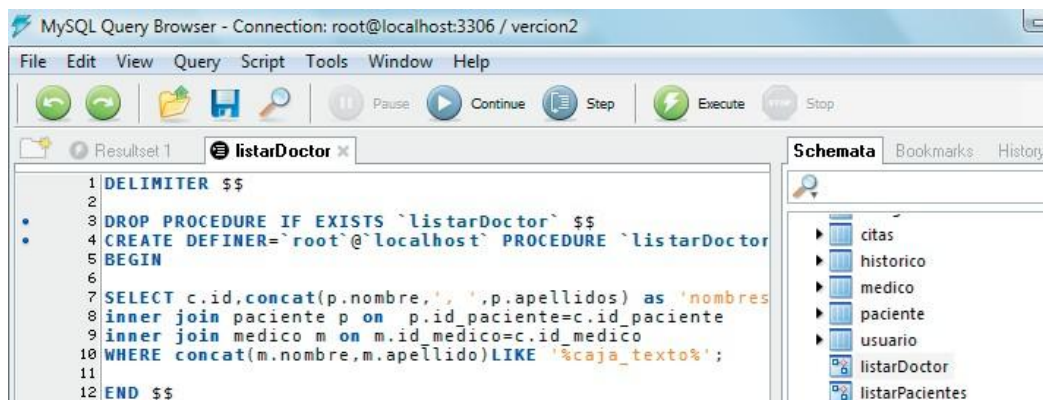


Diagrama 5. Base de datos MySQL.

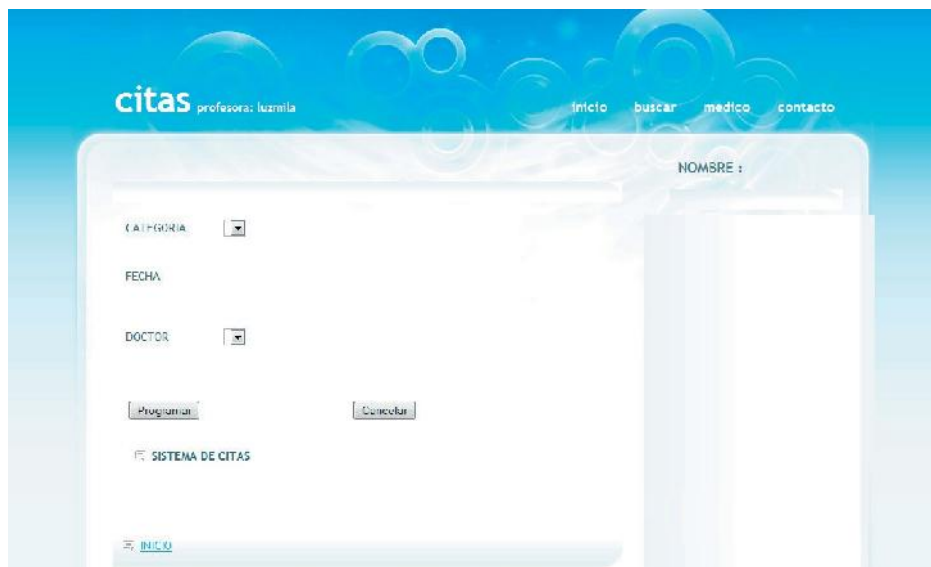


Figura 1. Asignación de cita al paciente

4.3. desarrollo de la Aplicación

El desarrollo del prototipo con JADE, se implementa en ide eclipse, la clase JADE, ubicado en el paquete agentes, controla con serlevt, se comunica con el cliente y la clase especial GatewayAgent, dba esta en la clase BD, user está en la clase beans.

Luego se ve el código para conectar a base de datos y el agente especial:

```
public class MiAgenteGateWay extends GatewayAgent {
```

En uno de sus métodos se emplea el tradicional acceso a la base de datos (ver la Figura 2).

El agente especial, conecta la tecnología JADE con otras tecnologías que son servlet, y para que funcione necesita de una clase importante que se le llamará:

```
JadeGateway.init("agentes.MiAgenteGateWay", null);
```

seguido de:

```
JadeGateway.execute(usuario);
```

El agente especial, accede la base de datos y se comunica entre agentes con el lenguaje ACL o puede comunicarse directamente con el servlet con el siguiente comando:

```
releaseCommand(usuario);
```

La bd usa las clases pacientes, doctor, gui; en tiempo de compilación y ejecución se transforman en agentes, depende del diseño su ciclo de vida, se inicia el agente GUI, (agente de middleware JADE). (7) En la Figura 3 los agentes DF y AMS, registran a cada agente. El df registra los servicios (páginas amarillas).

Se instala la plataforma JADE antes de correr la herramienta netbeans. Ver la Figura 4.

```
public String driver="com.mysql.jdbc.Driver";
public String url="jdbc:mysql://localhost:3306/analisis";
public String usuario="root";
public String clave="root";
public Connection getConnection(){
    Connection cn=null;
    try {
        Class.forName(driver).newInstance();
        cn=DriverManager.getConnection(url,usuario,clave);
```

Figura 2.

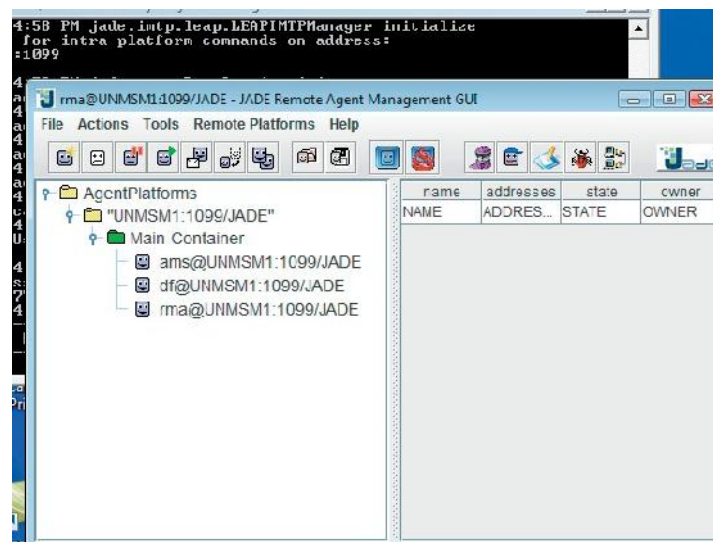


Figura 3.

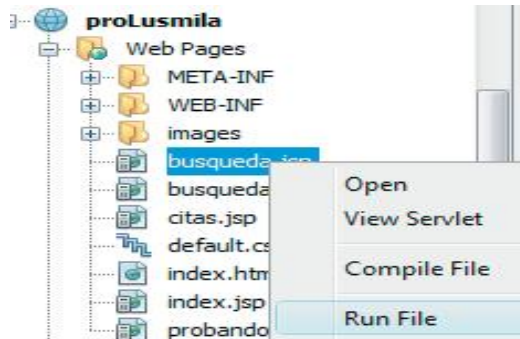


Figura 4

La Figura 5, muestra la creación de los agentes JADE como su creación.

El Objeto de Base De Datos es ::

com.mysql.jdbc.JDBC4ResultSet@18e1b3e

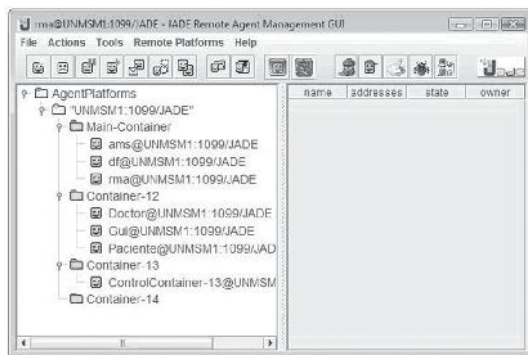


Figura 5.

Se utiliza sql para consultas, recuperar la data e impresión de la pantalla, el agente MiAgentGateway se encarga del acceso a la base de datos.

4.4. Modelo de diagnóstico de la tbc

El diagnóstico de la TBC se realiza con los pasos del proceso ontológico utilizando la herramienta Protégé. Ver en la Figura 6 los pasos 5 y 6.

5. CoNCLUSIoNeS

Del presente estudio se tiene las conclusiones siguientes:

1. Se presenta una base teórica de los Fundamentos de Ingeniería de la Web: Ontologías, Web Semántica y Agentes de Software .
2. Se presenta el análisis y el diseño del estudio se realizó en base al objetivo: apoyo en el diagnóstico de la TBC a través de Informes de Ministerio de Salud (MINSA), Se analizó la situación actual del servicio de emergencia de los Hospitales Dos de Mayo y Arzobispo Loayza en cuanto a la gestión citas en particular para el TBC.
3. Se presenta el Análisis y Diseño de un prototipo con sistema multiagentes para la gestión de citas y el apoyo en el diagnostico de la TBC.

6. ReFeReNCIAS bIbLIoGRÁFICAS

1. Botia, Juan 2005, *La Plataforma de Agentes JADE*. Univ. Sevilla.
2. Diseño AUMIL. 2004. Metodología para sistemas multiagentes.
3. FIPA Specification. Part 2, Agent Communications language, foundation for intelligent physical agents, Geneva, Switzerland. <http://www.cse.it.it/uvf/leonardo/fipa/index.htm>,1997.

```
// paso 5
(defrule Tiene tuberculosis(and (object (is-a paciente) (http://www.owl-ontologies.com/Ontology1290987510.owl#nombre
?ed))
(tbc (síntomas fatiga fiebre perdida _de_peso sudoración tos)))
=>(printout t "la persona" ?n "con una edad de " ?ed " tiene tuberculosis ".crlf))

// paso 6
(run)

// tiene que mostrar la inferencia:
Paciente de yyy años tiene tuberculosis
```

Figura 6. Fragmento de polar del procesor ontológico. Herramienta Protégé.

- [4]. GIRONDA, Cirilo. 1991, Programa Nacional de Control de la Tuberculosis, 1991.
- [5]. INGENIAS Development Kit <http://grasia.fdi.ucm.es/main/>
- [6]. <https://ingenias.svn.sourceforge.net/svnroot/ingenias/trunk/metamodel/metamodelGoPPRIlike.xml>
- [7]. JADE FIPA <http://jade.tilab.com/>
- [8]. The Linux home page at Linux online. 2004
- [9]. Mamani M. Néstor A. Criando uma Arquitetura de Memória Corporativa baseada em um Modelo de Negócio. <http://www-di.inf.puc-rio.br/~julio/tese-Nestor.pdf>
- [10] [http://www.minsa.gob.pe/portal/031/03Estrategias_Nacionales/04ESN-Tuberculosis\(Archivos\(Plan%20Estrategias%20TBC.doc](http://www.minsa.gob.pe/portal/031/03Estrategias_Nacionales/04ESN-Tuberculosis(Archivos(Plan%20Estrategias%20TBC.doc)
- [11].MySQL 5.0 Reference Manual, 2005.
- [12].PRÓ C. Luzmila, "Fundamentos de Ingeniería de la Web: Ontologías, Web Semántica y Agentes de Software", *Revista de Investigación de Sistemas e Informática (RISI)*, Volumen 7 Nº 1 año 2010, pp. de 77 a 89.
- [13].RUSSELL, N. 1996 *Sistemas Multiagentes*.
- [14].Tutorial de JSP Java Server Pages, 2004.
- [15].<http://www.eclipse.org/downloads/moreinfo/java.php/>
- [16].<http://www.fipa.org/specs/>
- [17].<http://www.who.int/topics/tuberculosis/es>.

