

**APLICACIÓN DE LA METODOLOGÍA DE DESARROLLO INGENIAS Y
TÉCNICAS DE WEB SEMÁNTICA EN LA IMPLEMENTACIÓN DE UN
CUADERNO ELECTRÓNICO DE INVESTIGACIONES**

**SARA LUCIA CASAS CASTAÑEDA
CARLOS ALBORNOZ BALAGUERA**

**UNIVERSIDAD AUTONOMA DE BUCARAMANGA
FACULTAD DE INGENIERIA DE SISTEMAS
INFORMACIÓN E INGENIERÍA DEL SOFTWARE
BUCARAMANGA**

2005

**APLICACIÓN DE LA METODOLOGÍA DE DESARROLLO INGENIAS Y
TÉCNICAS DE WEB SEMÁNTICA EN LA IMPLEMENTACIÓN DE UN
CUADERNO ELECTRÓNICO DE INVESTIGACIONES**

**SARA LUCIA CASAS CASTAÑEDA
CARLOS ALBORNOZ BALAGUERA**

**Trabajo de grado para optar al título de
Ingeniero de Sistemas**

**DIRECTOR:
ING. GARETH BARRERA SANABRIA**

**UNIVERSIDAD AUTONOMA DE BUCARAMANGA
FACULTAD DE INGENIERIA DE SISTEMAS
INFORMACIÓN E INGENIERÍA DEL SOFTWARE
BUCARAMANGA**

2005

Nota de Aceptación

Firma del Jurado

Firma del Jurado

Firma del Director

Bucaramanga, 15 de septiembre de 2005

AGRADECIMIENTOS

Queremos expresar nuestros más sinceros agradecimientos la Ingeniera Gareth Barrera Sanabria, nuestra directora, por toda su colaboración y orientación en el desarrollo de éste proyecto y más allá de eso, por creer en nosotros y hacernos partícipes de su confianza.

A la Universidad Autónoma de Bucaramanga, institución que nos abrió las puertas para hacernos profesionales.

A nuestras familias por habernos apoyado y servirnos de soporte hasta en los momentos más difíciles, no solo durante ésta etapa sino en todas las etapas vividas.

A nuestros amigos por su compañía, colaboración y comprensión.

A todos los que de una u otra forma estuvieron con nosotros, brindándonos su colaboración y apoyo.

DEDICATORIAS

A Dios y a mis padres por su inmenso cariño
y por el apoyo que siempre me han brindado,
a mis hermanas por su comprensión
y a Carlos Alberto por su amistad.

SARA LUCIA

A Dios, a mis padres y a mi hermana,
por el apoyo, la paciencia, la confianza
y el amor que siempre me han brindado.

A Sara Lucia por su amistad.

CARLOS ALBERTO

GLOSARIO

AGENTES: Wooldridge y Jennings definen un agente como “un programa auto-contenido capaz de controlar su proceso de toma de decisiones y de actuar, basado en la percepción de su ambiente, en persecución de uno o varios objetivos”

INGENIAS: Es una metodología de ingeniería de software orientada a agentes para el desarrollo de Sistemas Multi-Agente (SMA). Define un conjunto de entregas y actividades que ayudan a planear el desarrollo del sistema, ofreciéndole herramientas que facilitan la producción de esas entregas. Transmite conceptos de ingeniería tales como definición de flujos de trabajo, encapsulación de la funcionalidad a través de roles, objetivos, grupos y organizaciones

JAVA: lenguaje de objetos, □ independiente de la plataforma originalmente desarrollado por un grupo de ingenieros de Sun, su uso se destaca en el Web, sirve para crear todo tipo de aplicaciones (locales, Intranet o Internet).

JDOM: API (Application Programming Interface) pensada específicamente para el procesamiento de documentos XML con Java. Permite el parseado, creación, manipulación y serialización de documentos XML.

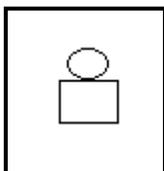
META-MODELO: Cada vista esta descrita en un meta-modelo, cada meta-modelo especifica una clase de gramática con la cual se construirán los modelos, conocidas como vistas. Describe que debe tener un SMA. Utilizando esos meta-modelos se identifican entidades que pueden aparecer en el futuro SMA.

SISTEMA MULTI-AGENTE (SMA): estudia la coordinación del comportamiento inteligente entre un grupo de agentes inteligentes autónomos. Se centra en el comportamiento individual a partir del cual se deriva el comportamiento del sistema. Se ocupa principalmente de estudiar modelos de comportamiento, estrategias de cooperación y coordinación, optimización del desempeño de tareas, aprendizaje a partir de experiencias propias, formación de coaliciones

CUADERNOS ELECTRÓNICOS: Es el equivalente a un Cuaderno de investigación, el cual está diseñado para guardar información de tipo texto, ecuaciones, imágenes, graficas, notas personales u otro tipo de datos mediante el computador o directamente desde instrumentos tecnológicos permitiendo ingresar información desde distintas ubicaciones geográficas, para hacer mas eficiente y ofrecer una mayor colaboración en el trabajo de equipo.

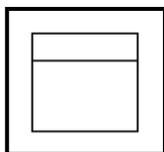
GLOSARIO INGENIAS IDE

AGENTE



Es una entidad autónoma con la identidad, propósitos y realiza las actividades para lograr sus metas.

APLICACIÓN



Una aplicación es un sistema computacional. Por “computacional” se entiende que tiene una interfaz y un comportamiento concreto.

Es usada cuando tenga alguna entidad del sistema (físico) que no se pueda categorizar como agente, organización, o recurso. Cuando no se saben desde el principio mas detalles de esa entidad.

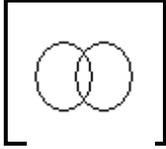
CREENCIA



Conjunto de afirmaciones que no son ciertas, sólo son suposiciones.

Se usa para definir lo que se espera de otros agentes

COMPROMISO



Entidad que expresa que un agente tiene que ejecutar una tarea debido a una petición realizada por otro agente

Usada para registrar la necesidad de ejecutar tareas debido a peticiones pasadas de otros agentes

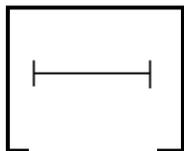
ESPECIFICACIÓN DE GRASIA



Descripción que permite hablar sobre la tecnología de acuerdo a las condiciones mentales que deben encontrarse a los iniciadores y colaboradores de cada paso, qué tareas se ejecutarán y cuando, y lo que es el orden de la ejecución de los actos de comunicación diferentes.

Use este tipo de diagramas en el plan generar una descripción exacta de qué tipo de interacción usted desea. En el análisis sólo uso los UML Colaboración diagramas.

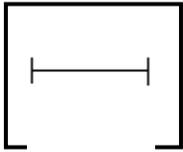
ESTADO MENTAL



Representa el estado mental de un agente en un determinado momento siendo un agregado de entidades mentales.

Se usa para expresar qué entidades mentales un agente supone para tener al principio a la activación. Esto se expresa en un modelo del agente asociando una entidad del agente a un estado mental.

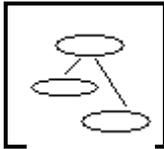
ESTADO MENTAL CONDICIONAL



Es un estado mental que agrega información extra sobre qué condiciones debe satisfacer las entidades agregadas en un estado mental. Las entidades en un estado mental se etiquetan. Estas etiquetas se usan dentro de la condición estado mental.

Es usado en la fase del plan, para determinar lo que usted requiere de cada entidad mental en un cierto momento.

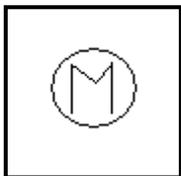
FLUJO DE TRABAJO



Un flujo de trabajo es una abstracción de un proceso que ha sido automatizado usando actividades e identificando sus responsables.

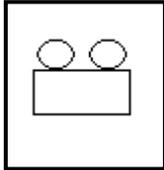
Los flujos de trabajo son usados para representar tareas ejecutadas en común por diferentes agentes o roles. La descripción de flujo de trabajo puede facilitarse con las interacciones para determinar en qué momento que se ejecutan las tareas.

GESTOR DEL ESTADO MENTAL



Es el estado mental en que el agente toma las decisiones. Este estado mental es un agregado de entidades mentales (creencias, hechos, eventos). La gerencia de estado mental consiste en determinar cómo se agregan las nuevas entidades mentales, cómo mantener la consistencia, y cómo quitar las entidades. Para especificar estos elementos usted puede usar campo de la descripción de la entidad o también la tarea y modelos de la meta. Si usted usa a la tarea y modelos de las metas, usted puede detallar qué tareas de dirección existen y cómo ellos actúan. Pueden asociarse las tareas a las entidades mentales por otras relaciones diferente de en estos diagramas, consume o produce.

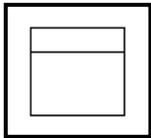
GRUPO



Un grupo puede contener otros grupos, roles, agentes, aplicaciones, o recursos. Un grupo representa la estructura de una organización.

Es usado para estructurar los elementos dentro de una organización. También, es útil cuando el diseñador supone un número alto de agentes que pueden estar trabajando juntos. Se puede comparar un Grupo como los departamentos de una empresa

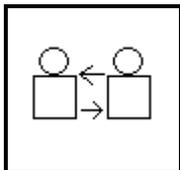
HECHO



Describe una información que el agente acepta como fiable. Esta entidad general contiene esta información en el campo de descripción.

Es usado para expresar una experiencia pasada del agente, la información pasada de tarea a tarea, y otra clase de información asumida como confiable para el agente.

INTERACCIÓN



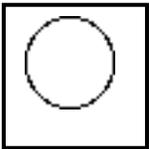
Representa una interacción entre dos o más agentes o roles. Puede haber sólo un iniciador y por lo menos un colaborador. Una interacción también detalla los objetivos que se persiguen. Este objetivo debe relacionarse con las metas de los participantes.

Es usado para indicar que hay una interacción y que tiene un propósito. En las fases avanzadas, esta entidad puede detallarse más allá asociando entidades de especificación especializadas en diferentes anotaciones, como diagramas de colaboración UML.

NOTA DEL TEXTO

Objeto gráfico que muestra el texto que explica detalles de un diagrama.
Se usa para clarificar algún aspecto de un diagrama

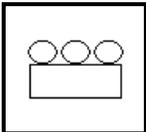
OBJETIVOS



Según el modelo de BDI, un objetivo es un estado deseado que un agente quiera alcanzar. Un objetivo es representado por un estado global, siendo una entidad por si misma, sin embargo puede relacionarse con una representación del estado global que usa las relaciones de satisfacción con las tareas. Estas relaciones contienen referencias a las descripciones de estados mentales de agentes, para que ellos se refieren a la imagen del mundo que tiene el agente.

Los objetivos son usados para representar los estados del mundo que quiera alcanzar. Esos estados pueden ser representados por conjuntos de entidades mentales asociados a estados mentales. Las tareas de los agentes deben proporcionar los medios para alcanzar los objetivos. También se tiene en cuenta que cualquier objetivo puede fallar. También hay relaciones que pueden representar este aspecto.

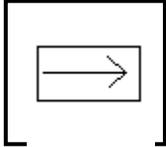
ORGANIZACIÓN



Una organización contiene un conjunto de agentes, roles y recursos que juntos consiguen alcanzar uno o varios objetivos. Una organización se descompone en grupos. Se puede relacionar una organización con una empresa. Internamente está compuesta por departamentos que pueden reestructurados sin afectar la imagen externa de una empresa.

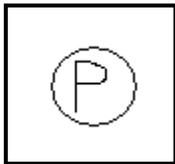
Use la organización como un grupo grande de elementos. Es recomendable usarlos cuando el número de agentes, roles o recursos se ponen difíciles manejar. Las organizaciones se refinan usando grupos.

PASO DE MENSAJE



Es la forma como se comunican los agentes. Se usa cada vez que dos o más agentes interactúan.

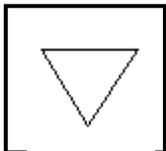
PROCESADOR ESTADO MENTAL



El agente toma las decisiones basandose en su estado mental. Hay una entidad que apoya gestion del estado mental del agente (Gestor del Estado Mental) y esta entidad que representa las capacidades de decisión del agente. Como el Gestor Estado Mental usted puede describir el Procesador Estado Mental usando diagramas de objetivos y tareas.

Se usa cuando se quiere hablar sobre los procedimientos de decisión del agente

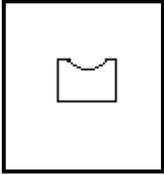
RECURSO



Un recurso facilita la ejecución de las tareas, hay recursos consumibles y no consumibles.

Use esta entidad cuando se quieran representar algunos requerimientos no funcionales, como por ejemplo, mantener el ancho de banda entre un valor y otro, o no usar mas del X % del tiempo de la CPU , o se necesita que por lo menos x archivos estén disponibles.

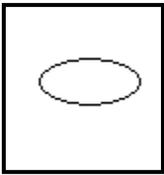
ROL



Un rol es una agrupación autónoma de funcionalidades. Cuando un agente juega un rol nosotros queremos expresar que el agente tiene que ejecutar tareas asociadas a ese rol y participar en las mismas interacciones del ese rol.

Use los roles cuando usted quiera buscar conjuntos de de tareas que pueden ser ejecutadas por diferentes tipos de agentes.

TAREA



Las tareas son una encapsulación de acciones o algoritmos no distribuibles. Las tareas pueden usar Aplicaciones y recursos. Las tareas generan cambios en el estado mental del agente que los ejecuta. Los cambios consisten en:

- Modificación, creación o destrucción de entidades mentales
- Los cambios en la percepción del mundo actuando sobre aplicaciones (las aplicaciones actúan sobre el mundo produciendo eventos, que son percibidos por el agente).

RESUMEN

En este proyecto se pretende dar solución al problema de administración de información de proyectos, usando una Nueva Metodología para el desarrollo de Sistemas Multiagentes, llamada INGENIAS. Esta metodología está basada en cinco meta-modelos (organización, agente, objetivos/tareas, e interacciones) ayudando a reflejar los resultados previos del análisis.

Este hecho permite trabajar directamente con los conceptos que intervienen en el desarrollo estructurando y detallando el diseño de un SMA que ofrezcan propiedades como autonomía, reactividad, proactividad, habilidad social y probarlos en casos reales. Por medio de ingenias se podrá satisfacer las necesidades del proyecto de Cuadernos Electrónicos, tales como la toma de anotaciones, búsqueda eficiente de información, trabajo en colaboración entre los investigadores, interoperabilidad, lenguaje de marcado para la generación de documentos o anotaciones y así facilitar el desarrollo del proyecto.

TABLA DE CONTENIDO

	Pág.
INTRODUCCION	
ANTECEDENTES	30
JUSTIFICACION	32
1. MARCO TEORICO	34
1.1 AGENTES	34
1.2 CLASIFICACIÓN DE AGENTES SOFTWARE	37
1.2.1 Agentes colaborativos	38
1.2.2 Agentes interfaz	38
1.2.3 Agentes móviles	38
1.2.4 Agentes de Información	38
1.2.5 Agentes Reactivos	39
1.2.6 Agentes Híbridos	39
1.2.7 Agentes Heterogéneos	39
1.2.8 Sistemas Multiagentes	39
1.3 ARQUITECTURAS DE AGENTES INTELIGENTES	40
1.3.1 Arquitecturas abstractas	40
1.3.1.1 Puramente reactivos	40
1.3.1.2 Perceptivos	40
1.3.1.3 Agentes con estados	41
1.3.2 Arquitecturas concretas	42
1.3.2.1 Basados en lógica	42

1.3.2.2	Arquitecturas reactivas	42
1.3.2.3	BDI (Comocimiento-deseo-inteligencia)	43
1.3.2.4	Basadas en capas	43
1.4	SISTEMAS MULTIAGENTE	45
1.4.1	Workflow y Gestión de Procesos de Negocio	45
1.4.2	Censores distribuidos controlados por Agentes	45
1.4.3	Agentes para comercio Electrónico	46
1.5	INGENIERÍA DEL SOFTWARE ORIENTADA A AGENTES ISOA	46
1.5.1	Relación entre agentes y objetos	47
1.5.2	Análisis y diseño orientado a agentes	48
1.5.3	Métodos formales para ISOA	48
1.5.4	Verificación formal	49
1.6	HERRAMIENTAS PARA EL DESARROLLO DE C.E	49
1.6.1	XML (eXtensible Markup Language)	49
1.6.1.1	Documentos XML	50
1.6.1.2	Etiquetas	51
1.6.2	DTD: Definición de Tipos de Documento	51
1.6.3	El modelo de objetos de documentos DOM	54
1.6.4	JDOM	55
1.6.4.1	Ventajas de JDOM	56
1.6.4.2	Desventajas de JDOM	56
1.6.5	JAVA	56
1.6.6	Java Server Pages (JSP)	57
2.	METODOLOGÍAS PARA DESARROLLO DE SISTEMAS MULTIAGENTES	59

2.1	METODOLOGIA GAIA PARA EL DESARROLLO DE SISTEMAS MULTIAGENTE	59
2.2	METODOLOGIA MESSAGE	60
2.3	METODOLOGÍA PARA EL DESARROLLO DE SISTEMAS MULTIAGENTE MAS – COMMONKADS	61
2.4	METODOLOGIA INGENIAS	62
2.4.1	Definicion del SMA con Ingenias	63
2.4.2	Meta-Modelo de Agente	64
2.4.2.1	Control del agente	67
2.4.3	Meta-Modelo de Objetivos y Tareas	69
2.4.3.1	Tareas en los Sistemas Multi-Agente	69
2.4.3.2	Objetivos en los Sistemas Multi-Agente	69
2.4.3.3	Presentación del meta-modelo de tareas y objetivos	70
2.4.4	Meta-Modelo de Organización	73
2.4.5	Meta-Modelo de Interacción	
	77ve en la figura	
2.4.5.1	Análisis y diseño de interacciones.	77
2.4.5.2	Elementos de la interacción	79
2.4.6	Meta-Modelo de Entorno	80
3.	CUADERNOS ELECTRONICOS	83
4.	FASE DE ANALISIS Y DISEÑO	90

4.1	DESCRIPCION DEL SISTEMA	90
4.2	METAMODELO DE AGENTES	94
4.2.1	Agente Investigador y Administración de Usuarios	94
4.2.2	Agente Cuaderno	96
4.2.3	Agente buscador y Agente G ⇒ Búsqueda	97
4.2.4	Agente Actualización	98
4.2.5	Agente Administración de Base de Datos	99
4.2.6	Agente Productos	99
4.2.7	Agente recordatorio	99
4.2.8	Agente asignación de permisos	99
4.3	META-MODELO DE ORGANIZACIÓN	103
4.4	META-MODELO DE INTERACCIÓN	106
4.4.1	Interacción Gestionar búsquedas	107
4.4.2	Interacción Administración e Investigador	107
4.4.3	Interacción gestión de permisos	108
4.4.4	Interacción de Gestión de recordatorios	109
4.4.5	Interacción actualización de C. E	109
4.5	META-MODELO DE OBJETIVOS Y TAREAS	110
4.6	METAMODELO DE ENTORNO	114
5.	FASE DE IMPLEMENTACION	118
5.1	HERRAMIENTAS NECESARIAS PARA LA IMPLEMENTACION	118
5.2	DESARROLLO CON XML	118

5.2.1	DTD Usuarios	119
5.2.2	DTD Cuaderno	119
5.2.3	DTD Anotación	120
5.2.4	DTD Proyectos	120
5.2.5	DTD Permisos de Usuarios	121
5.2.6	DTD Artículo	121
5.2.7	DTD Monografía	122
5.3	INTERFAZ GRAFICA	123
	CONCLUSIONES	132
	REFERENCIAS BIBLIOGRAFICAS	

LISTA I AS

	Pág
Figura 1. Un agente perceptivo.	41
Figura 2. Agente con estados.	42
Figura 3. Distintas arquitecturas basadas en capas	44
Figura 4. Ejemplo de la jerarquía de un Árbol XML	54
Figura 5. Esquema de funcionamiento de JSP	58
Figura 6. Relaciones entre los diferentes Meta-modelos y las dos entidades principales, la organización y el agente	64
Figura 7. Entidades principales en el Meta-modelo de agente	66
Figura 8. Conceptos relevantes en el control del agente	68
Figura 9. Relaciones entre objetivos y tareas	71
Figura 10. Formas en que una tarea afecta a un objetivo	72

Figura 11.	Vistas de una organización	74
Figura 12.	Meta-modelo de organización. Elementos que componen un flujo de trabajo	75
Figura 13.	Relaciones sociales	77
Figura 14.	Relación entre aspectos de la interacción	79
Figura 15.	Meta-modelo de entorno	80
Figura 16.	Meta-modelo de entorno. Tareas, recursos y aplicaciones	81
Figura 17.	Arquitectura propuesta para cuadernos electrónicos	88
Figura 18.	Agentes Administrador – Investigador	95
Figura 19.	Agente Cuaderno Electrónico	97
Figura 20.	Agentes Buscador y Gestión de búsqueda	98
Figura 21.	Agente Actualización	141
Figura 22.	Agente Administración de Base de Datos.	142
Figura 23.	Agente Producto	143
Figura 24.	Agente Recordatorio	144
Figura 25.	Agente Asignación de Permisos	145

Figura 26.	Meta-modelo de Organización	104
Figura 27.	Gestión de Búsquedas	107
Figura 28.	Interacción administración con Investigador	108
Figura 29.	Gestión de Permisos	146
Figura 30.	Actualización del Cuaderno Electrónico	147
Figura 31.	Gestión de Recordatorios	148
Figura 32.	Satisfacción de Objetivos	112
Figura 33.	Satisfacción de objetivos	146
Figura 34.	Flujo de Tareas	150
Figura 35.	Flujo de Tareas específico	151
Figura 36.	Recursos del Cuaderno Electrónico	116
Figura 37.	Página Principal del C.E	123
Figura 38.	Nuevo proyecto	124
Figura 39.	Agregar Anotación	125
Figura 40.	Nuevo Investigador	126

Figura 41.	Gestión de permisos	128
Figura 42.	Proyectos en desarrollo	129
Figura 43.	Creación de foros	130
Figura 44.	Creación de nuevos comentarios para foros	131

LISTA | BLAS

	Pág
Tabla 1. Instancias de entidades de meta-modelo de agentes	100
Tabla 2. Instancias de entidades de meta-modelo de Organización	105
Tabla 3. Instancias de entidades de meta-modelo de Interacción	109
Tabla 4. Instancias de entidades de meta-modelo de Objetivos y tareas	113
Tabla 5. Instancias de entidades de meta-modelo de Entorno	116

LISTA I NEXOS

	Pag
Anexo 1 Meta-modelo de agentes	141
Agente Actualización	
Anexo 2 Meta-modelo de agentes	142
Agente Admón. de Base de Datos.	
Anexo 3 Meta-modelo de agentes	143
Agente Producto	
Anexo 4 Meta-modelo de agentes	144
Agente Recordatorio	
Anexo 5 Meta-modelo de agentes	145
Agente Asignación de Permisos	
Anexo 6 Meta-modelo interacción	146
Gestión de Permisos	
Anexo 7 Meta-modelo interacción	147
Actualización del Cuaderno Electrónico	

Anexo 8 Meta-modelo interacción	148
Gestión de Recordatorios	
Anexo 9 Meta-modelo de objetivos y tareas	149
Satisfacción de objetivos	
Anexo 10 Meta-modelo de objetivos y tareas	150
Flujo de Tareas	
Anexo 11 Meta-modelo de objetivos y tareas	151
Flujo de Tareas específico	

INTRODUCCIÓN

La tecnología orientada a agentes ha sido uno de los paradigmas que durante los últimos años ha surgido en torno de los sistemas de desarrollo de software, y aunque en muchas oportunidades no se ha considerado como una tecnología totalmente nueva gracias a su semejanza con el paradigma orientado a objetos, está recibiendo una gran atención a nivel mundial.

Tradicionalmente, el registro de la información concerniente a los proyectos de investigación se lleva a cabo de forma manual, requiriendo un registro constante de datos que contengan resultados de pruebas, cuadros comparativos, figuras, planteamiento de objetivos entre otros.

Este esquema manual de registro impide un efectivo desempeño de un ambiente de colaboración y participación en proyectos de investigación; es por esta razón que el grupo investigador decidió diseñar una herramienta de tipo interno de investigación para la administración de información de los proyectos realizados por el Grupo de Tecnologías de Información del Laboratorio de Computo Especializado de la Universidad Autónoma de Bucaramanga. Con el desarrollo de dicha herramienta se busca facilitar la toma de anotaciones y el trabajo en colaboración con otros investigadores. Para lograr el objetivo propuesto, se llevará a cabo la aplicación de la metodología de desarrollo de Sistemas Multiagentes INGENIAS¹, la cual nos proporciona un enfoque más estructurado y detallado

¹ GÓMEZ, J. Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. No.18, pp. 51-63. ISSN: 1137-3601. © AEPIA (<http://www.aepia.dsic.upv.es/>). Metodologías para el desarrollo de Sistemas multi-agente Jorge J. Gómez Sanz 2003

para la creación de agentes inteligentes, que nos permitan realizar procesos automáticos más eficientes por medio de sus cinco Meta-modelos. Los Meta-modelos de INGENIAS son: Meta-modelo de Agente (describe Agentes basándose en su funcionalidad), Meta-modelo de Objetivos y Tareas² (recoge motivaciones del SMA, define acciones y cómo afectan tales acciones), Meta-modelo de Organización (arquitectura del Sistema en un SMA), Meta-Modelo de Interacción (identifica dependencias entre componentes) y Meta-modelo de Entorno (define qué existe alrededor del Sistema y cómo lo percibe cada agente).

En este documento se describen las diferentes metodologías orientadas a agentes y que desventajas tienen sobre la metodología INGENIAS con sus respectivos meta-modelos en el desarrollo de Cuadernos Electrónicos. Además se mostrará el diseño del cuaderno electrónico desarrollado por los investigadores.

² CONNOLLY, F. van Harmelen, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. DAML+OIL Reference Description. W3C. Disponible en <http://www.w3.org/TR/daml+oil-reference> Note 18 December 2001

ANTECEDENTES

El prototipo de Cuadernos Electrónicos desarrollado por los investigadores, es de tipo interno de investigación con el fin de dar solución a la forma de tomar y buscar anotaciones de proyectos. Inicialmente, se desea implantar en el Centro de Cómputo Especializado de la Universidad Autónoma de Bucaramanga con miras a ser implementado en otras facultades o para uso general de estudiantes.

Para el desarrollo de este CE se quiere implementar el uso de Agentes para facilitar la toma de decisiones usando la Metodología INGENIAS para el desarrollo de Sistemas Multiagentes. Actualmente existen en la UNAB varios proyectos relacionados con la Tecnología de Agentes. El primero fue realizado por las Ingenieras Gareth Barrera y Carolina Rodríguez con título “Aplicación de una metodología orientada a agentes en la implantación de un Sistema de reserva automáticas de vuelos”³ y el segundo fue realizado por Maria Clemencia Montagú y Jorge Leonardo Vargas que tiene como título “Aplicación de la Metodología ingenias en la implementación de un prototipo de software para la administración de una cadena de suministros (supply-chain management)”⁴. El tercero fue realizado por Norma Judith Casas y Diana Quintanilla con el título “Diseño e implementación de un prototipo de comercio electrónico utilizando un paradigma

³ BARRERA Sanabria Gareth, Rodríguez Buitrago Carolina. Aplicación de una metodología orientada a agentes en la implantación de un sistema de reserva de vuelos. Universidad Autónoma de Bucaramanga. Facultad de Ingeniería de Sistemas, 2001

⁴ MONTAGÚ Castro, María Clemencia, Vargas Mayorga, Jorge Leonardo. Tesis Aplicación de la metodología ingenias en la implementación de un prototipo de Supply Chain Manageme. Universidad Autónoma de Bucaramanga. Facultad de Ingeniería de Sistemas, 2004

orientado a agentes”⁵, el cuarto fue realizado por Juan Carlos García con el título “Buscadores inteligentes de información basados en la tecnología de agentes móviles” ⁶, el quinto fue realizado por José Fabián Díaz y Anderson Murillo titulado “Diseño e implementación de un prototipo de mercado virtual utilizando la tecnología de agentes” ⁷ y el sexto fue realizado por Eduardo Martínez, William Prieto y Freddy Pico , titulado “Prototipo de aplicación de comercio electrónico utilizando la metodología Gaia al desarrollo de Sistemas Multiagentes” ⁸.

Referenciados en el segundo proyecto anteriormente mencionado, el proyecto de Cuadernos Electrónicos busca implementar una herramienta de Sistemas Multiagentes haciendo un estudio detallado de las metodologías existentes. También requiere el uso de Lenguajes de Mercado como XML, el cuál hace posible definir los recursos, propiedades y tareas de una forma mas estructurada.

⁵ CASAS Castañeda Norma Judith, Quintanilla Diana Patricia. Tesis Diseño e implementación de un prototipo de comercio electrónico utilizando un paradigma orientado a agentes. Universidad Autónoma de Bucaramanga. Facultad de Ingeniería de Sistemas, 2002.

⁶ GARCÍA Juan Carlos. Buscadores inteligentes de información basados en la tecnología de agentes móviles. Universidad Autónoma de Bucaramanga. Facultad de Ingeniería de Sistemas, 2001

⁷ DÍAZ Silva José Fabián, Murillo Anderson. Diseño e implementación de un prototipo de mercado virtual utilizando la tecnología de agentes. Universidad Autónoma de Bucaramanga. Facultad de Ingeniería de Sistemas, 2003

⁸ MARTINEZ Eduardo, Prieto William y Freddy Pico. Prototipo de aplicación de comercio electrónico utilizando la metodología Gaia al desarrollo de software orientado a agentes. Universidad Autónoma de Bucaramanga. Facultad de Ingeniería de Sistemas, 2002.

JUSTIFICACION

Tradicionalmente, el trabajo en investigación requiere un registro constante de datos (resultados de pruebas, cuadros comparativos, figuras, planteamiento de objetivos), registro que se lleva a cabo generalmente en papel. Este esquema de registro, impide un efectivo desarrollo de elementos como ambientes de colaboración con participación en proyectos de investigación y procesos de búsqueda eficientes, que enriquezcan los procesos investigativos.

Los Cuadernos Electrónicos son una herramienta utilizada en diversas áreas de investigación ⁹, los cuales son una buena opción para compartir información. La metodología de Cuadernos Electrónicos se convertido en una revolucionaria forma de tomar anotaciones sobre investigación y experimentos si se trabaja en un ambiente de colaboración y sincronización. La idea de un Cuaderno electrónico es eliminar las hojas y los cuadernos corrientes de las investigaciones, cumpliéndose a cabalidad con las funciones normales tales como resultado de pruebas, cuadros comparativos, figuras, planteamiento de objetivos, además se mejoran otros procesos como el registro automático de las fechas de cada anotación y búsqueda de información a través del cuaderno los cuales crea confianza, eficacia y presteza en el proceso de investigación.

Países más avanzados como Italia, Alemania, Estados Unidos, Japón, Canadá, México, Australia y Suiza emplean técnicas similares a CE de uso privado que les

⁹ GEIST, AI Design of The DOE2000 Electronic Notebook : The Electronic Notebook Architecture. Berkeley California. December 1997

permite acceder a la información confidencial desde cualquier lugar con equipos cuyo Sistema operativo es diferente, usando la Internet para obtener resultados óptimos en el menor tiempo posible¹⁰. Colombia, aunque en vía de desarrollo, no es ajena al uso de Paradigmas Orientados a Agentes para cuadernos electrónicos que le permita ponerse a la vanguardia en el uso de Sistemas Multiagente que ofrezcan propiedades como autonomía, reactividad, proactividad y habilidad social.

En el Laboratorio de Cómputo Especializado de la Universidad Autónoma de Bucaramanga, no se ha diseñado una herramienta que facilite la toma de anotaciones y que ayude al progreso de las investigaciones. Tampoco se ha observado la existencia de un ambiente de colaboración en línea, ya que es muy difícil compartir los resultados y progresos de sus investigaciones. En consecuencia el grupo investigador ha sugerido este proyecto con el fin de diseñar e implementar un cuaderno electrónico que permita agilizar y optimizar la toma de notas tipo texto de las investigaciones y el ambiente de trabajo entre los participantes ofreciendo seguridad por medio de permisos de acceso a los investigadores participantes, interoperabilidad permitiendo el uso del cuaderno electrónico desde cualquier unidad de trabajo con diverso Sistema operativo, búsquedas de información en el CE usando el lenguaje de marcado para crear búsquedas más exactas y trabajo en colaboración por medio de foros establecidos por cada investigador.

El desarrollo de un cuaderno electrónico que involucre características del paradigma orientado a agentes y anotaciones de documentos con lenguajes de marcado, propiciaría un mejor aprovechamiento de las notas recogidas en el desarrollo de un proyecto de investigación, brindando un espacio para el trabajo en colaboración entre investigadores con intereses en un proyecto en común, facilitando además, una consulta inteligente sobre los contenidos del mismo.

¹⁰ D'INVERNO, Mark y Michael Luck, 3-540-41975-6, Springer, Understanding Agent Systems, Marzo 2001

1. MARCO TEORICO

1.1 AGENTES

No existe una definición exacta para el término Agente, debido a su utilización en diferentes áreas de la ciencia de computación. Según Wooldridge, un agente es un programa de autocontenido que es capaz de controlar sus acciones y decisiones para alcanzar unos objetivos, basándose en su percepción del entorno¹¹, que puede realizar alguna acción definida para un usuario, con un nivel de inteligencia que le permite desarrollarla de manera autónoma y que puede interactuar con su entorno, siendo la definición mas acertada.

Sin embargo no se descartan enunciaciones tales como la de Russell¹² refiriéndose a que un agente es algo que puede ser visto percibiendo el entorno y actuando sobre él ó una entidad software con un propósito específico, o bien a la apreciación hecha por IBM¹³ en la que hace referencia a entidades software que llevan a término un conjunto de operaciones bajo las órdenes de un usuario o de otro programa, con un cierto grado de independencia o autonomía, empleando conocimiento o representaciones de los objetivos del usuario.

¹¹ WOOLDRIDGE and N. R. Jennings. Agent theories, architectures, and languages: A survey. In M. Wooldridge and N. R. Jennings, editors, Intelligent Agents: Theories, Architectures, and Languages (LNAI Volume 890), Springer-Verlag: Heidelberg, Germany, Jan. 1995.

¹² RAO, A y M. Georgeff. Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference, Morgan Kaufmann, San Mateo, 1-55860-262-3, Editado por B. Nebel, C. Rich, y W. Swartout, "An abstract architecture for rational agents, 1992

¹³ IBM Agent Building and Learning Environment (ABLE). [online] Available from World Wide Web: <<http://www.alphaworks.ibm.com/tech/able>> [cited 15 february 2004]

Entre los investigadores que más han contribuido en el tema de agentes, se encuentran Wooldridge y Jennings, quienes definen dos nociones para el término Agente: noción débil y noción fuerte.

Noción débil de agencia. Es un Sistema hardware o Software¹⁴ basado en computador con las siguientes propiedades:

- Autonomía: Los agentes han de actuar sin la intervención directa de los humanos o de otros agentes y además han de tener algún tipo de control sobre sus acciones y su estado interno¹⁵. Toma la iniciativa y ejerce el control sobre sus propias acciones en la siguiente forma:
 - Orientado a objetivo: Actúa en función de los requerimientos de alto nivel y decide cómo y dónde satisfacerlos.
 - Colaborativo: No obedece ciegamente los comandos debido a que puede modificar las solicitudes, realizar preguntas de clarificación, o rehusar satisfacer ciertas solicitudes.
 - Flexible: Es capaz de escoger dinámicamente que acciones invocar, y en qué secuencia, en respuesta al estado de su entorno.
 - Auto iniciable: Puede sentir los cambios de su entorno y decidir cuando actuar.
- Habilidad social: Interactúan con otros Agentes por un lenguaje de comunicación con otros agentes incluyendo personas, para obtener información o conseguir ayuda para llevar a cabo sus objetivos.
- Reactividad: Los Agentes perciben su entorno, y responden.
- Proactividad: Exhiben un comportamiento dirigido por objetivos para tomar la iniciativa actuando solo en respuesta al entorno.

¹⁴ GEIST, AI Design of The DOE2000 Electronic Notebook : The Electronic Notebook Architecture. Berkeley California. December 1997

¹⁵ CONNOLLY, F. van Harmelen, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. DAML+OIL Reference Description. W3C Note 18 December 2001. Disponible en <http://www.w3.org/TR/daml+oil-reference>

- Continuidad temporal: Significa que está continuamente corriendo, no es un programa que transforma entradas simples en salidas simples y luego termina.
- Personalidad: Tiene una personalidad creíble, bien definida, que facilita la interacción con usuarios humanos.
- Adaptabilidad: Automáticamente se adapta a los cambios en su entorno y a las preferencias de su usuario, basándose en la experiencia previa.

La noción de agencia definida anteriormente es la utilizada por la Ingeniería del Software Orientada a Agentes: “Los agentes se comunican con otros por el intercambio de mensajes en un lenguaje de comunicación de agente expresivo. Mientras los agentes pueden ser simples subrutinas, ellos son generalmente entidades grandes con un tipo de control persistente”¹⁶.

Noción fuerte de agencia: Un agente es un Sistema basado en computador¹⁷ que además de autónomo, proactivo, reactivo y con habilidad social usa nociones mentales como conocimiento, creencias, intenciones y obligaciones, además de movilidad (Puede transportarse de una máquina a otra y a través de diferentes arquitecturas y plataformas), veracidad¹⁸ (un agente no comunicará conscientemente información falsa), racionalidad¹⁹ (un agente actuará de acuerdo con sus objetivos, y no lo hará de forma que evite que sus objetivos sean cumplidos como mínimo, hasta donde sus creencias se lo permitan).

¹⁶ TIMBERNERS-LEE, J. Hendler, O Lassila. Fascinating facts about Tim Berners-Lee inventor of the World Wide Web. The Semantic Web 12-589-6587-AK25, <http://www.ideafinder.com/history/inventors/berners-lee.htm> Scientific American, May 2001

¹⁷ WOOLDRIDGE, Michael et al. Agent – oriented software engineering for Internet applications. An introduction to Multiagent Systems. June 2000

¹⁸ ROSENSCHEIN, J and GENESERETH, M. Deals among rational agents. En : Proceedings of the ninth International joint conference on artificial intelligent. October 1985

¹⁹ GALLIERS, J. A theoretical framework for computer models of cooperative dialogue, acknowledging multiagent conflict. Tesis (PhD). Open University Uk. Citado por : IGLESIAS FERNÁNDEZ, Carlos Ángel. Definición de una metodología para el desarrollo de sistemas multiagente. Tesis (Doctoral). Universidad Politécnica de Madrid. Departamento de Ingeniería de Sistemas Telemáticos. España, Julio 1998, 321 p.

Comunicación entre Agentes: Las soluciones al problema de comunicación en los MAS puede ir desde los Sistemas²⁰ que no se comunican hasta los que mantiene comunicación:

- No hay comunicación: Los agentes han de inferir los planes de los demás sin establecer ningún tipo de comunicación.
- Primitivas de comunicación: La comunicación está restringida a un número de señales fijas con una interpretación establecida de antemano. Esta limita la cooperación entre agentes.
- Intercambio de planes e información: Dos agentes pueden intercambiarse sus respectivos planes, y cada uno puede adaptar sus estrategias: alto coste computacional del intercambio, y no se garantiza el plan resultante.
- Intercambio por medio de un tablero: es una manera de compartir memoria y conocimiento. Los agentes pueden escribir mensajes, dejar resultados parciales o encontrar información en un tablero que todos saben donde está.
- Intercambio de mensajes: agentes que actúan en respuesta al procesamiento de una comunicación. Las acciones que pueden ejecutar estos agentes son: enviarse una comunicación a ellos mismos, crear otros agentes del mismo tipo, y especificar el reemplazo del comportamiento.
- Comunicación en alto nivel: Diálogo entre agentes donde el emisor pretende llevar al receptor al mismo estado mental que el emisor.

Para que los Agentes puedan interactuar e intercambiar información y servicios deben comunicarse por medio de un lenguaje de comunicación de Agentes.

- Transporte: cómo envían y reciben mensajes los Agentes
- Lenguaje: qué significan los mensajes
- Política: estructura de conversaciones entre Agentes

²⁰ WOOLDRIDGE, Michael, 0-471-49691-X, John Wiley & Sons, An introduction to Multiagent Systems. June 2000

- Arquitectura: cómo conectar Sistemas.

1.2 CLASIFICACIÓN DE AGENTES SOFTWARE

A continuación se presentan una clasificación de agentes según Hyacinth²¹.

1.2.1 Agentes colaborativos. Se fundamentan en la autonomía y la cooperación (con otros agentes), de manera que facilite que estos realicen sus tareas al igual que él lleva a cabo las suyas. Este tipo de agentes pueden aprender, pero este aspecto no es en el que su operación hace más énfasis.

1.2.2 Agentes interfaz. La labor de este tipo de agente se fundamenta en la autonomía y el aprendizaje para facilitar el trabajo de los demás agentes (humanos). Pattie Maes²² una de las promotoras de esta clase de agentes, expone que la principal tarea de los agentes interfaz es ser un asistente personal que está colaborando con el usuario en el mismo ambiente de trabajo. Existe una sutil diferencia entre colaborar con el usuario y colaborar con otros agentes. Colaborar con el usuario puede no requerir un lenguaje de comunicación de agente explícito como se requiere cuando se colabora con otros agentes.

1.2.3 Agentes móviles. Existen diferentes definiciones para el término agente móvil. Un agente móvil es un agente que no permanece en la plataforma de agente donde comienza su ejecución y puede transportarse subsecuentemente por sí mismo entre diferentes plataformas de agentes. La tecnología de agentes móviles, proporciona ventajas tales como costos de comunicación reducidos, recursos locales limitados, facilidad de coordinación, computación asíncrona, además de proporcionar una arquitectura computacional distribuida.

²¹ IBM Agent Building and Learning Environment (ABLE). [online] Available from World Wide Web: <<http://www.alphaworks.ibm.com/tech/able>> [cited 15 february 2004]

²² WOOLDRIDGE, Michael, 2002, 0-471-49691-X, John Wiley & Sons, *An introduction to Multiagent Systems*

1.2.4 Agentes de información. Los agentes de información se han desarrollado gracias a la clara demanda de herramientas para las herramientas para manejar el explosivo crecimiento de la información que se está experimentando actualmente y el cual seguiremos experimentando. Los agentes de información llevan a cabo un rol asociado con el manejo, manipulación o comparación de información de diversas fuentes distribuidas.

1.2.5 Agentes reactivos. Este tipo de agentes representan una categoría especial de agentes los cuales no poseen modelos internos, simbólicos del entorno; en cambio ellos actúan / responden de manera simultánea al estado presente del entorno en el cual ellos están inmersos.

1.2.6 Agentes híbridos. Los agentes híbridos se refieren a aquellos cuya constitución es una combinación de dos o más filosofías de agente dentro de un solo agente. Estas filosofías incluyen una filosofía móvil, una filosofía de interfaz, una filosofía de agente en colaboración. La hipótesis clave de tener agentes o arquitecturas híbridas es la creencia de que para alguna aplicación, los beneficios van a ser mejores que hacer uso de una filosofía singular.

1.2.7 Agentes heterogéneos. Los Sistemas de agentes heterogéneos, a diferencia de los agentes híbridos, se refieren a un grupo integrado de cómo mínimo dos agentes, los cuales pertenecen a dos o más clases de agentes diferentes. Un Sistema de agente heterogéneo puede además contener uno o más agentes híbridos.

1.2.8 Sistemas Multiagente. Un agente Internet es una entidad software autónoma, la cual interactúa con su entorno (Internet) y con otros agentes proactivamente (esto es, por iniciativa propia) de manera que pueda llevar a cabo sus propios objetivos.

Los Sistemas Multiagente pueden ser considerados como agrupaciones de agentes autónomos, actuando y trabajando independientemente el uno del otro. Cada agente trata de acoplar su propio entorno de manera que pueda acceder a información / servicios que no posee o coordinar sus actividades para asegurar que sus objetivos pueden ser cumplidos.

1.3 ARQUITECTURA DE AGENTES INTELIGENTES

1.3.1 Arquitecturas abstractas²³

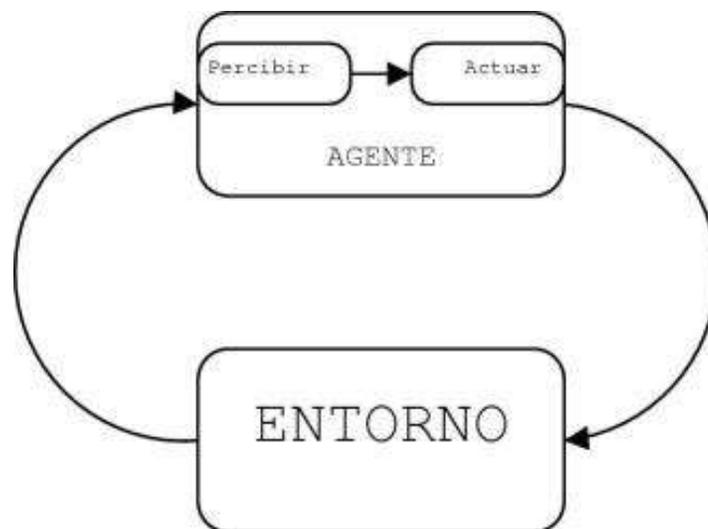
1.3.1.1 Puramente reactivos. Algunos tipos de agentes deciden que hacer sin ningún tipo de referencia a su historia. Estos basan sus decisiones completamente en el presente, sin ninguna referencia al pasado. Se pueden llamar puramente reactivos ya que responden directamente al entorno.

1.3.1.2 Perceptivos. Desde un punto de vista abstracto de los agentes es posible realizar un análisis sencillo, pero esto no nos ayuda a construirlos. Si comenzamos a profundizar en lo que sería el funcionamiento del agente nos encontramos diseñando sus subSistemas de funcionamiento. Sin llegar todavía a una arquitectura en concreto podemos diferenciar dos subSistemas percepción y acción.

La idea es que la función de percepción engloba la habilidad del agente de observar su entorno mientras que la acción representa el proceso de toma de decisiones como se muestra en la figura 1.

²³ CASTELFRANCHI, C. Guarantees for autonomy. En : Cognitive Agent Architecture. (1995). Citado por : IGLESIAS FERNÁNDEZ, Carlos Ángel. Definición de una metodología para el desarrollo de sistemas multiagente. España, 1998, 321 p. Tesis (Doctoral). Universidad Politécnica de Madrid. Departamento de Ingeniería de Sistemas Telemáticos

Figura 1. Un agente perceptivo



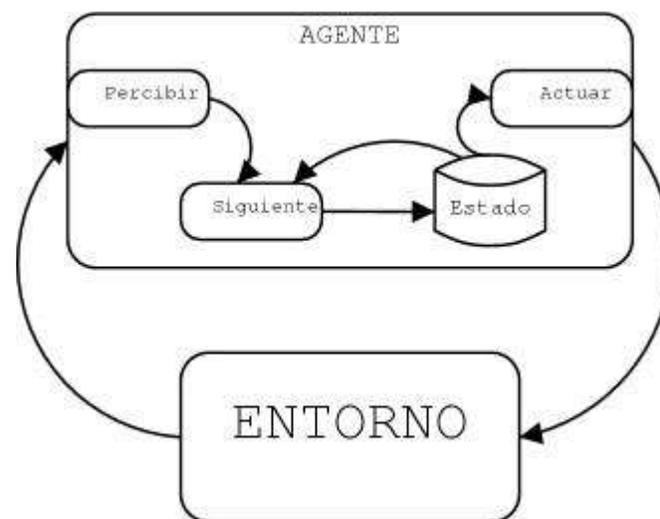
Fuente. GUTTMAN, R. H. y A. G. Moukas *The Knowledge Engineering Review*, Cambridge University Press, 0269-8889, Editado por Simons Parsons y Adele E. Howe, "Agent-mediated electronic commerce: a survey", 1998, 147-159

1.3.1.3 Agentes con estados. Refinando más el modelado del agente como se muestra en la figura 2, podemos almacenar una historia de estados del entorno o percepciones y acciones realizadas. De esta forma los agentes podrían poseer una estructura interna usada para registrar información sobre el entorno y acciones realizadas.

El comportamiento de un agente basado en estados se puede resumir como sigue. El agente comienza en un determinado estado. Observa su entorno

generando una percepción. El estado interno del agente es actualizado. La acción seleccionada por el agente se realiza y el agente entra en otro ciclo percibiendo el mundo, actualizando su estado y seleccionando una nueva acción.

Figura 2. Agente con estados



Fuente. WOOLDRIDGE, Michael et al. 0-471-49691-X, Agent – oriented software engineering for Internet applications. An introduction to Multiagent Systems October 2002.

1.3.2 Arquitecturas concretas

1.3.2.1 Basadas en lógica. La aproximación tradicional para construir agentes inteligentes sugiere que el comportamiento inteligente puede ser generado en un Sistema dando una representación simbólica del entorno y un comportamiento deseado. La idea es que el programador codifique las reglas deductivas y la base de conocimiento de forma que al derivar se seleccione la mejor acción posible.

1.3.2.2 Arquitecturas reactivas. La arquitectura reactiva mas conocida es sin lugar a duda la arquitectura de desarrollada por Rodney Brooks (MIT)²⁴, caracterizada por aproximarse a conductas de alto nivel (inteligentes) adoptando como modelos, organismos vivientes de bajo nivel (por ejemplo insectos), y desarrollando así, entidades (robots) que evolucionarán en su comportamiento, emulando patrones de conducta presentados por diversos organismos vivientes.

Esta arquitectura proporciona capacidades a través de la implementación de entidades simples con controles descentralizados, representaciones no compartidas, y algunas otras características más.

Brooks concibe que la aproximación hacia la Inteligencia Humana en las máquinas, puede ser guiada por los mecanismos y fenómenos que se presentan en la evolución de las especies. Afirma que se deben concebir primero inteligencias simples y gradualmente intentar la concepción de conductas más complejas²⁵.

1.3.2.3 BDI (Conocimiento-deseo-intención). En una arquitectura BDI, el estado del agente se representa mediante tres estructuras: su conocimiento, sus deseos y sus intenciones²⁶. El conocimiento de un agente son su modelo de entorno, sus deseos proporcionan cierto orden entre los estados y las intenciones son las cosas que decide hacer. Las intenciones de uno de estos agentes pueden ser definidas desde varios niveles de abstracción; por ejemplo, un agente puede intentar comprar un libro en particular, pero no ha decidido todavía en que librería lo

²⁴ GUTTMAN, R. H. y A. G. Moukas *The Knowledge Engineering Review*, Cambridge University Press, 0269-8889, Editado por Simons Parsons y Adele E. Howe, "Agent-mediated electronic commerce: a survey", 1998, 147-159

²⁵ JACOBSON, I., Booch, G. y Rumbaugh, J.: *El Proceso Unificado de Desarrollo de Software*. Libro completo. Addison Wesley. 303-3792000. January 2001

²⁶ O. LASSILA, O., R. R. Swick. Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation 22 February 1999. Available at <http://www.w3.org/TR/REC-rdf-syntax>

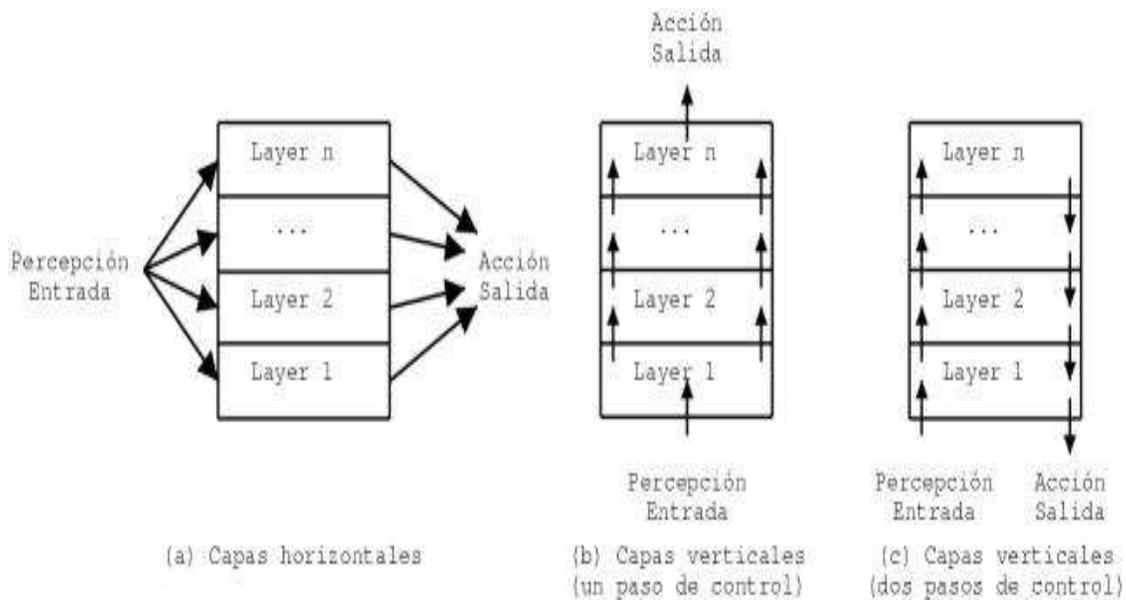
comprará. Un agente BDI redefine gradualmente sus intenciones para poder ejecutar acciones primitivas.

1.3.2.4 Basadas en capas. Una forma muy extendida de abordar la implementación de un agente es enfrentar su diseño en capas, tal como se hace por ejemplo en las redes de computadoras.

La idea es la de que distintas capas, cada una encargada de un determinado proceso, cooperan entre ellas para llevar a cabo una acción como lo muestra la figura 3. De esta forma se encuentra capas que pueden trabajar paralelamente para cumplir un objetivo u organizarse verticalmente proporcionando servicio unas a otras.

Como se ve en la figura 3, en una arquitectura horizontal todas las capas tienen acceso a sensores y actuadores, ofreciendo la ventaja del paralelismo entre capas a costa de un alto conocimiento de control para coordinarlas. Por su parte en las arquitecturas verticales se reduce este control a costa de una mayor complejidad en la capa que interactúa con los sensores.

Figura 3. Distintas arquitecturas basadas en capas



Fuente: WOOLDRIDGE, Michael et al. 0-471-49691-X, Agent – oriented software engineering for Internet applications. An introduction to Multiagent Systems October 2002.

1.4 SISTEMAS MULTIAGENTE

"Un Sistema Multiagente es aquel que contiene una colección de dos o mas agentes" ²⁷.

Dado que los agentes trabajan para cumplir objetivos pero son incapaces de generar sus propios objetivos, un Sistema Multiagente no tendría razón de ser sin un agente autónomo que genere estos objetivos. Por tanto se añade a la definición la necesidad de que exista como mínimo un agente autónomo dentro del Sistema. Por último se pide que exista una interacción entre los agentes del Sistema, ya que de lo contrario tendríamos un conjunto de agentes actuando

²⁷ IBM Agent Building and Learning Environment (ABLE). <http://www.alphaworks.ibm.com/tech/able> [cited 15 february 2004]

[online] Available from World Wide Web:

independientemente de los demás, lo que los convertiría en una colección de componentes y no en un Sistema.

Los agentes han encontrado aplicación en múltiples dominios²⁸. A continuación se hará un rápido repaso entre aplicaciones que hacen uso de Sistemas Multiagente.

1.4.1 Workflow y Gestión de Procesos de Negocio. Es un área de creciente importancia en las ciencias informáticas. Estos Sistemas ayudan a automatizar los procesos de negocio, asegurando que diferentes tareas son ejecutadas, además de asegurar que el flujo de documentos se gestiona correctamente dentro de la organización.

1.4.2 Censores distribuidos controlados por agentes. La aplicación clásica de la tecnología Multiagente fue en los sensores distribuidos. La idea principal es tener un Sistema construido como una red de sensores distribuidos espacialmente para monitorizar por ejemplo todos los vehículos que pasen dentro del rango de los sensores. Esta tarea es más sencilla si los sensores en la red cooperan entre ellos, intercambiando predicciones sobre cuando un vehículo pasara de una región de un sensor a la de otro.

1.4.3 Agentes para Comercio Electrónico. Con el boom de Internet a partir de finales de los 90s creció el interés en el comercio electrónico. El problema reside en que la Web tiene orígenes académicos, siendo diseñada para un libre acceso. Esto la hace en un principio poco eficiente para ser usada como "mercado" debido a temas de: Privacidad, seguridad, confianza, pago entre otras

Para mejorar los Sistemas de venta empezaron a crearse Sistemas Multiagente que simulaban el comportamiento de los consumidores a la hora de comprar. La

²⁸ JACOBSON, I., Booch, G. y Rumbaugh, J.: El Proceso Unificado de Desarrollo de Software. Libro completo. Addison Wesley. 303-3792000. January 2001

Universidad de Cambridge²⁹ postulo que los consumidores tendían a realizar en el mayor número de casos, pasos como identificación, búsqueda del mejor producto, búsqueda del mejor proveedor, negociación y recepción del producto y evaluación. Los agentes son capaces de automatizar total o parcialmente algunas de estas etapas, ayudando al comprador a alcanzar el mejor trato posible.

1.5 INGENIERÍA DEL SOFTWARE ORIENTADA A AGENTES ISOA

La tecnología de agentes hace un gran aporte al campo de la ingeniería de software, no solo porque se muestra como un paradigma que busca mejorar las falencias en el desarrollo de software, sino por la creación de metodologías y herramientas para el desarrollo de agentes, en donde los elementos presentados en la metodología se integran con un paradigma de ingeniería del software; dicho paradigma asegura la calidad del desarrollo estableciendo métricas y aportando medios para gestionar su evolución³⁰.

1.5.1 Relación entre agentes y objetos. A los programas familiares orientados a objetos le falta tener nuevas ideas de agentes. Los objetos están definidos como entidades computacionales que encapsulan algún estado, y son capaces de realizar acciones o métodos permitiendo comunicarse entre ellos por medio de mensajes.

Los objetos y los agentes tienen notables distinciones:

- El grado en el que los agentes y los objetos son autónomos (llamando a la programación orientada a objetos principio de encapsulación). En lenguajes

²⁹ GÓMEZ, J. Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. No.18, pp. 51-63. ISSN: 1137-3601. © AEPIA (<http://www.aepia.dsic.upv.es/>). Metodologías para el desarrollo de Sistemas multi-agente Jorge J. Gómez Sanz 2003

³⁰ WOOLDRIDGE and N. R. Jennings. Agent theories, architectures, and languages: A survey. In M. Wooldridge and N. R. Jennings, editors, Intelligent Agents: Theories, Architectures, and Languages (LNAI Volume 890), Springer-Verlag: Heidelberg, Germany, Jan. 1995

de programación como Java, se puede declarar variables de instancia (en métodos) *private*, haciendo que ellas sean accesibles únicamente por otros objetos (o *public* permitiendo el acceso de cualquiera, hasta de otros objetos; cuando se usa este tipo de asignación, es usual considerar pobre el estilo de la programación). Muchos tipos de Sistemas Multiagentes (en particular los agentes que son construidos para organizaciones) no buscan dar solución a lo que es asignado³¹.

- El comportamiento autónomo flexible (reactivo, proactivo y social). Este estándar del modelo de objetos dice que no cualquiera puede decir acerca de cómo construir Sistemas e integrar su comportamiento.
- Los agentes consideran tener el control, están continuamente activos, y están típicamente comprometidos a observar el ambiente, evaluando el estado interno, seleccionando y ejecutando acciones de acuerdo a su estructura³².

Los objetos están quietos por mucho tiempo y llegan a ser activos únicamente cuando otro objeto requiere de sus servicios invocando un método. Un objeto activo es generalmente autónomo, puede exhibir su comportamiento y opera con otros objetos; mientras que un objeto pasivo, puede únicamente realizar su labor y cambiar de estado cuando se necesite. Los objetos activos son esencialmente agentes que no necesariamente tienen la habilidad de exhibir su comportamiento autónomo flexible.

1.5.2 Análisis y diseño orientado a agentes. Una aproximación en el desarrollo de Sistemas de agentes es involucrarse principalmente en metodologías de desarrollo informal para el análisis y diseño de Sistemas basados en agentes.

³¹ TVEIT, AMUND; A survey of Agent-Oriented Software Engineering, Norwegian University of Science and Technology, May 2001. Page 156 - 203

³² TVEIT, AMUND; A survey of Agent-Oriented Software Engineering, Norwegian University of Science and Technology, May 2001 Page 236 - 290

Estos Sistemas pueden dividirse en dos grandes grupos: Los que se inspiran en el desarrollo orientado a objetos y se extienden hasta metodológicas orientadas a objetos o que adaptan una de ellas; y los que se adaptan a la ingeniería del conocimiento o a otras técnicas.

1.5.3 Métodos formales para ISOA (Ingeniería de Software Orientada a Agentes)

Métodos Formales en especificación: Las técnicas predominantes para especificación de los Agentes se han adoptado como Sistemas intencionales que pueden ser entendidos atribuyéndolo a los estados mentales, como lo son convicciones, deseos e intenciones. Siguiendo esta idea, un número de técnicas para formalizar especificaciones de Agentes puede ser desarrollado ³³, y ser capaces de representar aspectos de un Sistema basado en Agentes tales como, las convicciones de estos agentes tienen (la información sobre su ambiente que puede ser incompleto o incorrecto), los objetivos que pueden ser conseguidos, las acciones de Agentes ejecutados, los efectos de estas acciones, y la interacción que está haciéndose sobre agentes.

Métodos formales en implementación: La especificación no es usualmente el fin de la historia en el desarrollo de software. Se proponen algunas formas abstractas de especificación para modelos computacionales concretos.

1.5.4 Verificación formal. Este proceso es conocido como verificación y es particularmente importante si tenemos que introducir alguna informalidad dentro del proceso de desarrollo.

La técnica de verificación de Sistemas se puede dividir en dos amplias clases:

³³ TVEIT, AMUND; A survey of Agent-Oriented Software Engineering, Norwegian University of Science and Technology, May 2001. Page 145 - 153

Técnicas Axiomáticas: Verificación Deductiva. Las técnicas axiomáticas para verificación de programas fueron las primeras en entrar a la ciencia de la computación y requiere poder tomar un programa concreto, y desde este programa derivar sistemáticamente una teoría lógica que representa su comportamiento. Este es llamado el programa teoría.

Técnica Semántica: Modelo de Verificación. Hasta hoy la técnica de axionomía para verificación es limitada inherentemente de la dificultad de la prueba de este problema. La adición de conectores temporales y modales para la lógica, consideran éste como un problema difícil. Esta técnica esta basada en la semántica del lenguaje de especificación.

1.6 HERRAMIENTAS PARA EL DESARROLLO DE AGENTES

1.6.1 XML (eXtensible Markup Language). Según su historia La versión 1.0 del lenguaje XML es una recomendación del W3C³⁴ desde Febrero de 1998, pero se ha trabajado en ella desde un par de años antes. Está basado en el anterior estándar SGML (Standard Generalized Markup Language, ISO 8879), que data de 1986, pero que empezó a gestarse desde principios de los años 70, y a su vez basado en el GML creado por IBM³⁵ en 1969. Esto significa que aunque XML pueda parecer moderno, sus conceptos están más que asentados y aceptados de forma amplia. Está además asociado a la recomendación del W3C DOM (Document Object Model)³⁶, aprobado también en 1998. Éste no es más que un modelo de objetos (en forma de API) que permite acceder a las diferentes partes que pueden componer un documento XML o HTML.

SGML proporciona un método consistente y preciso de aplicar etiquetas para describir las partes que componen un documento, permitiendo el intercambio de documentos entre diferentes plataformas. Manteniendo su misma filosofía, de él se derivó XML como subconjunto simplificado, eliminando las partes más engorrosas y menos útiles.

1.6.1.1 Documentos XML. Existen dos tipos de documentos XML: válidos y bien formados. Éste es uno de los aspectos más importantes del lenguaje:

Bien formados: son todos los que cumplen las especificaciones sintácticas del lenguaje, sin estar sujetos a unos elementos fijados en un DTD (definición de tipo de Documento). De hecho los documentos XML deben tener una estructura jerárquica muy estricta y los documentos bien formados deben cumplirla.

³⁴ WOOLDIDGE and N. R. Jennings. Agent theories, architectures, and languages: A survey. In M. Wooldridge and N. R. Jennings, editors, *Intelligent Agents: Theories, Architectures, and Languages* (LNAI Volume 890), Springer-Verlag: Heidelberg, Germany, Jan. 1995.

³⁵ HYACINTH, Nwana. Software agents: An overview [online]. Disponible en: <http://labs.bt.com/proyectos/agents/publish/papers/review.html> [cited 25 august 2004]

³⁶ WOOLDRIDGE, Michael, 2002, 0-471-49691-X, John Wiley & Sons, *An introduction to Multiagent Systems*

Válidos: Además de estar bien formados, siguen una estructura y una semántica determinada por un DTD donde sus elementos y la estructura jerárquica que se define deben ajustarse a lo que el DTD dicte.³⁷

1.6.1.2 Etiquetas. Las etiquetas son las que nos recuerdan a HTML, y que contienen los datos. Es importante diferenciar entre elementos y etiquetas: los elementos son las entidades en sí, lo que tiene contenido, mientras que las etiquetas sólo describen a los elementos. Un documento XML está compuesto por elementos, y en su sintaxis éstos se nombran mediante etiquetas.

Hay dos tipos de elementos: los vacíos y los no vacíos. Hay varias consideraciones importantes a tener en cuenta al respecto:

- Toda etiqueta no vacía debe tener una etiqueta de cerrado: `<etiqueta>` debe estar seguida de `</etiqueta>`. Esto se hace para evitar errores de interpretación.
- Todos los elementos deben estar perfectamente anidados (`<ficha><nombre>Angel</nombre> </ficha>`).
- Los elementos vacíos son aquellos que no tienen contenido dentro del documento. Un ejemplo en HTML son las imágenes. La sintaxis correcta para estos elementos implica que la etiqueta tenga siempre esta forma: `<etiqueta/>`.

³⁷ JAVA SERVLET & JSP Cookbook By Bruce W. Perry. Publisher : O'Reilly Pub Date : January 2004 ISBN : 0-596-00572-5 Pages : 746 - 793

1.6.2 DTD (Definición de Tipos de Documento). Los documentos XML pueden ser válidos o bien formados. En cuanto a los válidos, ya sabemos que su gramática está definida en los DTD³⁸. Los DTD no son más que definiciones de los elementos que puede incluir un documento XML, de la forma en que deben hacerlo (qué elementos van dentro de otros) y los atributos que se les puede dar.

Existen varios modos de referenciar un DTD en un documento XML:

Incluir dentro del documento una referencia al documento DTD en forma de URI (Universal Resource Identifier, o identificador universal de recursos) y mediante la siguiente sintaxis:

```
<!DOCTYPE ficha SYSTEM
"http://www.dat.etsit.upm.es/~abarbero/DTD/ficha.dtd">
```

En este caso la palabra SYSTEM indica que el DTD se obtendrá a partir de un elemento externo al documento e indicado por el URI que lo sigue, por supuesto entrecomillado.

O bien incluir dentro del propio documento el DTD de este modo:

```
<?xml version="1.0"?>
<!DOCTYPE ficha [
<!ELEMENT ficha (nombre+, apellido+, direccion+,
foto?)>
<!ELEMENT nombre (#PCDATA)>
<!ATTLIST nombre sexo (masculino|femenino) #IMPLIED>
<!ELEMENT apellido (#PCDATA)>
<!ELEMENT direccion (#PCDATA)>
<!ELEMENT foto EMPTY>
]>
```

³⁸ FERBER, J. y Gutknecht, O.: A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems. Actas de conferencia. Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS98), IEEE CS Press. 1998

```
<ficha>
<nombre>Angel</nombre>
<apellido>Barbero</apellido>
<direccion>c/Ulises, 36</direccion>
</ficha>
```

La forma de incluir el DTD directamente como en este ejemplo pasa por añadir a la declaración <!DOCTYPE y después del nombre del nombre del tipo de documento, en vez de la URI del DTD, el propio DTD entre los símbolos '[' y ']'. Todo lo que hay entre ellos será considerado parte del DTD.

A la hora de indicar los elementos descendientes vemos que van seguidos de unos caracteres especiales: '+', '*', '?' y '|'. Sirven para indicar qué tipo de uso se permite hacer de esos elementos dentro del documento:

- + : uso obligatorio y múltiple; permite uno o más elementos de ese tipo dentro del elemento padre, pero como mínimo uno.
- * : opcional y múltiple; puede no haber ninguna ocurrencia, una o varias.
- ? : opcional pero singular; puede no haber ninguno o como mucho uno.
- | : equivale a un OR, es decir, da la opción de usar un elemento de entre los que forman la expresión, y solo uno.

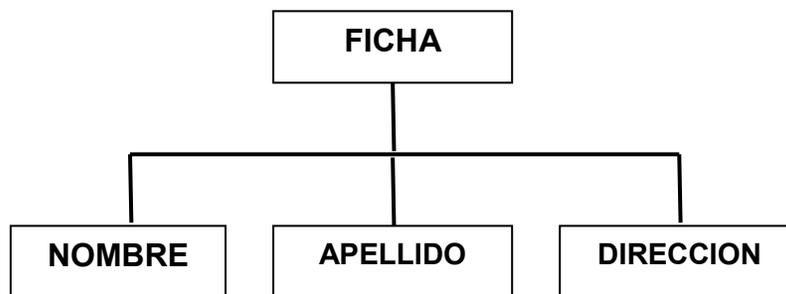
De este modo, si por ejemplo encontramos en un DTD la siguiente declaración:

```
<!ELEMENT ficha (nombre+, apellido+, direccion*, foto?, telefono*|fax*)>
```

Sabremos del elemento ficha que puede contener los siguientes elementos: un nombre y un apellido como mínimo, pero puede tener más de uno de cada; opcionalmente puede incluirse una o varias direcciones, pero no es obligatorio; opcionalmente también se puede incluir una única foto; y por fin, pueden incluirse, aunque no es obligatorio en ninguno de los dos casos, uno o más teléfonos o uno o más números de fax.

Un documento XML presenta una jerarquía muy determinada, definida en el DTD si es un documento válido, pero siempre inherente al documento en cualquier caso, con lo que se puede representar como un árbol de elementos. Existe un elemento raíz, que siempre debe ser único (como se muestra en la figura 4) y que se llamará como el nombre que se ponga en la definición del <!DOCTYPE si está asociado a un DTD o cualquiera que se desee en caso contrario. Y de él descienden las ramas de sus respectivos elementos descendientes o hijos. De este modo, la representación en forma de árbol de nuestro documento XML de ejemplo sería:

Figura 4. Ejemplo de la jerarquía de un Árbol XML



Fuente: FERBER, J. y Gutknecht, O.: A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems. Actas de conferencia. Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS), IEEE CS Press. 1998

El DTD, por ser precisamente la definición de esa jerarquía, describe precisamente la forma de ese árbol. La diferencia está en que el DTD define la forma del árbol de elementos, y un documento XML válido puede basarse en ella para estructurarse. Un documento XML bien formado sólo tendrá que tener una estructura jerarquizada, pero sin tener que ajustarse a ningún DTD concreto.

1.6.3 El modelo de objetos de documentos DOM. El modelo de objetos de documentos del W3Consortium, o Document Object Model (DOM) es una

representación interna estándar de la estructura de un documento, y proporciona una interfaz al programador (API) para poder acceder de forma fácil, consistente y homogénea a sus elementos, atributos y estilo. Es un modelo independiente de la plataforma y del lenguaje de programación.

El W3C establece varios niveles de actuación, coincidiendo con el tiempo en que se presentan como recomendación:

- **Nivel 1:** se refiere a la parte interna, y modelos para HTML y XML. Contiene funcionalidades para la navegación y manipulación de documentos. Tiene 2 partes: el coreo parte básica, referida a documentos XML, y la parte HTML, referida precisamente a los HTML.
- **Nivel 2:** incluye un modelo de objetos e interfaz de acceso a las características de estilo del documento, definiendo funcionalidades para manipular la información sobre el estilo del documento. También incluirá un modelo de eventos para soportar los XML *namespaces* y consultas enriquecidas.
- Posteriores niveles especificarán interfaces a posibles Sistemas de ventanas, manipulación de DTD y modelos de seguridad.

El objetivo es que de una vez por todas cualquier script pueda ejecutarse de forma más o menos homogénea en cualquier navegador que soporte dicho DOM. Siempre por supuesto se podrá elegir el implementar modelos propietarios que es lo que ahora ofrecen Netscape y Microsoft, pero tener una plataforma estándar en la que poder crear contenidos sin temor a no estar soportado por alguna marca o versión de navegador, que además sea potente y versátil.

1.6.4 JDOM (Java Document Object Model). Inventado por Brett McLaughlin y Jason Hunter, junto con la colaboración de James Duncan Davidson, bajo la licencia de apache-like, en la primavera del 2000. Se diseño estrictamente para

XML, totalmente en java y sin ninguna preocupación por tener compatibilidad con API's similares.

JDOM es una API (Application Programming Interface) pensada específicamente para el procesamiento de documentos XML con Java. Al igual que DOM (document Object Model) se basa en el parseado de un documento XML y la construcción de un árbol de Elementos, Atributos, Comentarios, Instrucciones de procesamiento, etc. Una vez construido el árbol se puede acceder directamente a cualquiera de sus componentes. A diferencia de DOM, que está pensada para ser una API genérica, implementable con cualquier lenguaje de programación, JDOM está expresamente pensada para Java, así pues los Elementos, Atributos, etc, se representarán mediante clases Java, sin que exista el concepto de Nodo propio de DOM.

1.6.4.1 Ventajas de JDOM:

- Pensada especialmente para Java y por tanto mejor integrada en este lenguaje.
- Facilidad de uso (para programadores que conozcan Java)
- Representación de documentos como árboles, lo que implica el acceso aleatorio a cualquier parte del documento (ventaja sobre SAX)
- API de libre distribución.

1.6.4.2 Desventajas de JDOM:

- Necesario procesar y almacenar en memoria todo el documento. Mayor consumo de memoria que SAX.
- Al ser una API diseñada para Java resulta poco portable a otros lenguajes de programación.

1.6.5 JAVA. Lenguaje originalmente desarrollado por un grupo de ingenieros de Sun, utilizado por Netscape posteriormente como base para Javascript. Si bien

su uso se destaca en la Web, sirve para crear todo tipo de aplicaciones (locales, Intranet o Internet)³⁹. Java es un lenguaje de objetos, independiente de la plataforma. Se caracteriza por ser robusto, gestiona la memoria automáticamente, no permite el uso de técnicas de programación inadecuadas, cliente-servidor, mecanismos de seguridad incorporados y herramientas de documentación incorporadas.

Además de incorporar la ejecución como **Applet**, **Java** permite fácilmente el desarrollo tanto de arquitecturas cliente-servidor como de aplicaciones distribuidas, consistentes en crear aplicaciones capaces de conectarse a otros ordenadores y ejecutar tareas en varios ordenadores simultáneamente, repartiendo por lo tanto el trabajo. Aunque también otros lenguajes de programación permiten crear aplicaciones de este tipo, **Java** incorpora en su propio **API** estas funcionalidades.

1.6.6 Java Server Pages (JSP). Es la tecnología para generar páginas web de forma dinámica en el servidor, desarrollado por Sun Microsystems, basado en scripts que utilizan una variante del lenguaje java⁴⁰. La principal ventaja de JSP frente a otros lenguajes es que permite integrarse con clases Java (class) lo que permite separar en niveles las aplicaciones web, almacenando en clases java las partes que consumen más recursos así como las que requieren más seguridad, y dejando la parte encargada de formatear el documento Html en el archivo Jsp.

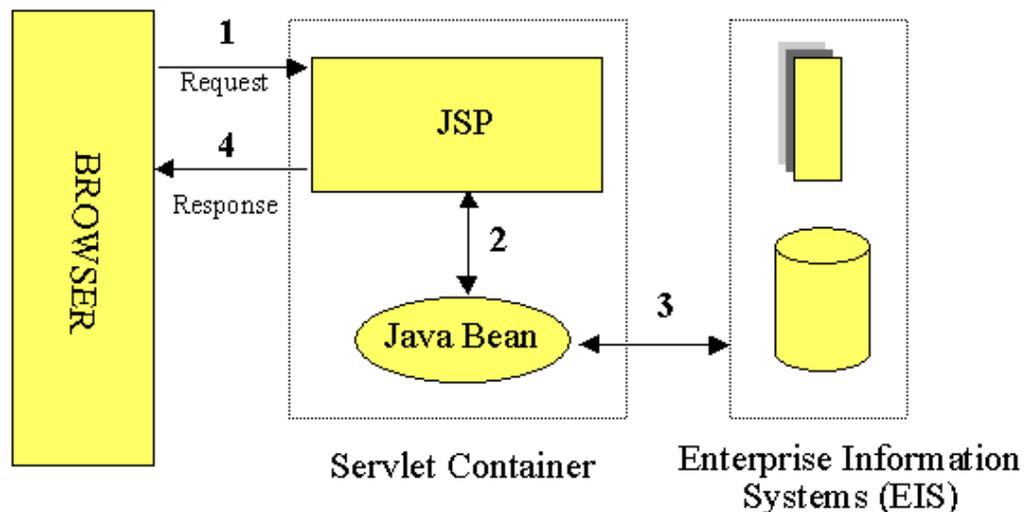
Los *servlets* y *Java Server Pages (JSPs)* son dos métodos de creación de páginas web dinámicas en servidor usando el lenguaje Java. Para empezar, los JSPs y servlets se ejecutan en una máquina virtual Java, lo cual permite que se puedan usar en cualquier tipo de ordenador, siempre que exista una máquina virtual Java

³⁹ MYERS, James D. Elena S. Mendoza , Bonnie Hoopes. A collaborative electronic laboratory notebook , Pacific Northwest, National Laboratory PO Box 999 Richland, WA 99352 USA

⁴⁰ JAVA SERVLET & JSP Cookbook By Bruce W. Perry. Publisher : O'Reilly Pub Date : January 2004 ISBN : 0-596-00572-5 Pages : 746 - 783

para él. Cada servlet (o JSP, a partir de ahora lo usaremos de forma indistinta) se ejecuta en su propia hebra, es decir, en su propio contexto; pero no se comienza a ejecutar cada vez que recibe una petición, sino que persiste de una petición a la siguiente, de forma que no se pierde tiempo en invocarlo (cargar programa + intérprete). Los JSPs son en realidad servlets: un JSP se compila a un programa en Java la primera vez que se invoca, y del programa en Java se crea una clase que se empieza a ejecutar en el servidor como un servlet.

Figura 5. Esquema de funcionamiento de JSP



Fuente. JAVA SERVLET & JSP Cookbook By Bruce W. Perry. Publisher : O'Reilly Pub Date : January 2004 ISBN : 0-596-00572-5

La principal diferencia entre los servlets y los JSPs (como se muestra en la figura 5) es el enfoque de la programación: un JSP es una página Web con etiquetas especiales y código Java incrustado, mientras que un servlet es un programa que recibe peticiones y genera a partir de ellas una página web.

2. METODOLOGÍAS PARA DESARROLLO DE SISTEMAS MULTIAGENTES

El paradigma orientado a agentes ha despertado un gran interés a nivel mundial, lo que ha conllevado al desarrollo de múltiples herramientas y metodologías para el desarrollo de sistemas multiagente. En la actualidad existen diversas metodologías orientadas a agentes, de las cuales se estudian algunas de ellas: metodología GAIA para desarrollo de sistemas multiagente, Metodología Message y la metodología MAS - CommonKADS.

El principal objetivo de esta comparación, es destacar las principales características de la metodología Ingenias, tomando como punto de partida algunas metodologías que han surgido recientemente para el desarrollo de sistemas multiagente, para poder determinar claramente las ventajas y desventajas que dicha metodología posee ante las otras metodologías existentes.

2.1 METODOLOGÍA GAIA PARA EL DESARROLLO DE SISTEMAS MULTIAGENTE

La metodología GAIA, permite realizar un análisis sistemático que abarca la definición de los requerimientos del sistema, que facilita la elaboración de un diseño lo suficientemente detallado, de manera que pueda ser implementado directamente⁴¹. La etapa de análisis empieza desde un punto de vista abstracto, el cual se descompone secuencialmente, hasta poder definir conceptos (clases, entidades) concretos. El proceso de análisis y diseño se describen a continuación respectivamente.

- Proceso de análisis: el proceso de análisis posee las siguientes actividades: modelo de roles (se identifican los papeles claves en el sistema), modelo de interacción (dependencias y relaciones entre varios papeles en una organización multiagente), identificación de protocolos, y la elaboración del proceso de roles.

- Proceso de diseño: el proceso de diseño tiene los siguientes modelos: modelo de agente (documenta los diferentes tipos de agentes, y las instancias de los mismos), modelo de servicio (en este modelo se identifican los servicios asociados con cada papel de los agentes, y se especifican las propiedades principales de esos servicios), y el modelo de comunicación (define la comunicación que existe entre diferentes tipos de agentes).

La metodología GAIA se aplicó inicialmente en un sistema basado en agentes para el manejo de procesos de negocios, revelando su utilidad y consistencia en el desarrollo de sistemas multiagente.

2.2 METODOLOGIA MESSAGE (Methodology for Engineering Systems of Software Agents).

Ésta es una metodología orientada a agentes la cual incorpora técnicas de ingeniería del software cubriendo el análisis y diseño de sistemas multiagente.

⁴¹ WOOLDRIDGE, Michael, 2002, 0-471-49691-X, John Wiley & Sons, *An introduction to Multiagent Systems*

La metodología provee un lenguaje, un método y unas guías de cómo aplicar la metodología, centrándose en las fases de análisis y diseño y lanzando ideas sobre el resto de etapas como implementación, pruebas e implantación⁴².

Esta además es reciente y trata de integrar resultados de las anteriores metodologías, propone cinco puntos de vista para capturar los diferentes aspectos de un SMA: vista de la organización , que captura la estructura global del sistema; vista de tareas y objetivos, que determina los objetivos que persiguen y las tareas que están implicadas en el proceso; vista de agente, contiene la información detallada de cada agente y el papel que cumple este dentro del Sistema; vista de Dominio que recolecta información concerniente con el problema y por ultimo la vista de interacción , que trata las interacciones a distintos niveles de abstracción.

Ésta es una metodología orientada a agentes la cual incorpora técnicas de ingeniería del software cubriendo el análisis y diseño de sistemas Multiagente. MESSAGE⁴³ provee un lenguaje, un método y unas guías de cómo aplicar la metodología, centrándose en las fases de análisis y diseño y lanzando ideas sobre el resto de etapas como implementación, pruebas e implantación.

2.3 METODOLOGÍA PARA EL DESARROLLO DE SISTEMAS MULTIAGENTE MAS – COMMONKADS

La propuesta que se presenta en este proyecto se fundamenta en la metodología MAS - CommonKADS, la cual presenta una estructura clara y completa para el desarrollo de sistemas multiagente. Esta metodología fusiona las tecnologías

⁴² EURESCOM. MESSAGE: Methodology for engineering systems of software agents. Initial methodology. Technical Report P907-D1, EURESCOM

⁴³ EURESCOM (2001b). MESSAGE: Methodology for engineering systems of software agents (Final). Technical Report P907-TI1, EURESCOM

KADS (*Knowledge Acquisition and Documentation System*) y la orientada a objetos para facilitar su aplicación⁴⁴.

MAS – CommonKADS se apropia de las características de las metodologías que no modelan el conocimiento (estudiadas anteriormente), y agrega a su conjunto de modelos un modelo de la experiencia, que permite realizar una estructura y representación completa del conocimiento de los agentes que intervienen en el sistema. Dada esta característica tan especial de MAS – CommonKADS fue seleccionada como la metodología a estudiar y a aplicar en el desarrollo de este proyecto⁴⁵.

La metodología MAS - CommonKADS se fundamenta básicamente en siete modelos, que tienen como objetivo permitir diseñar un método que cubra todos los aspectos trascendentales en el desarrollo de sistemas multiagente.

2.4 METODOLOGIA INGENIAS

La Metodología INGENIAS surge de una línea de trabajo cuyo propósito es consolidar la experiencia adquirida de la Metodología MESSAGE. Esta Metodología proporciona medios para construir Sistemas Multi-Agente (SMA) de forma disciplinada y repetible ⁴⁶.

La construcción de SMA integra tecnologías de distintas áreas de conocimiento: técnicas de ingeniería del Software para estructurar el proceso de desarrollo,

⁴⁴ IGLESIAS FERNÁNDEZ, Carlos Ángel *et al.* Analysis and design of multiagent systems using MAS – CommonKADS. Valladolid, España. (1999); Page 15 to 19

⁴⁵ IGLESIAS FERNÁNDEZ, Carlos Ángel *et al.* Analysis and design of multiagent systems using MAS – CommonKADS. Valladolid, España. (1999); Page 23 to 45

⁴⁶ GÓMEZ, J. Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. No.18, pp. 51-63. ISSN: 1137-3601. © AEPIA (<http://www.aepia.dsic.upv.es/>). Metodologías para el desarrollo de Sistemas multi-agente Jorge J. Gómez Sanz 2003

técnicas de inteligencia artificial para dotar a los programas de capacidad para tratar situaciones imprevistas y tomar decisiones, y programación concurrente y distribuida para tratar la coordinación de tareas ejecutadas en diferentes máquinas bajo diferentes políticas de planificación.

INGENIAS permite desarrollar un SMA mediante tres elementos:

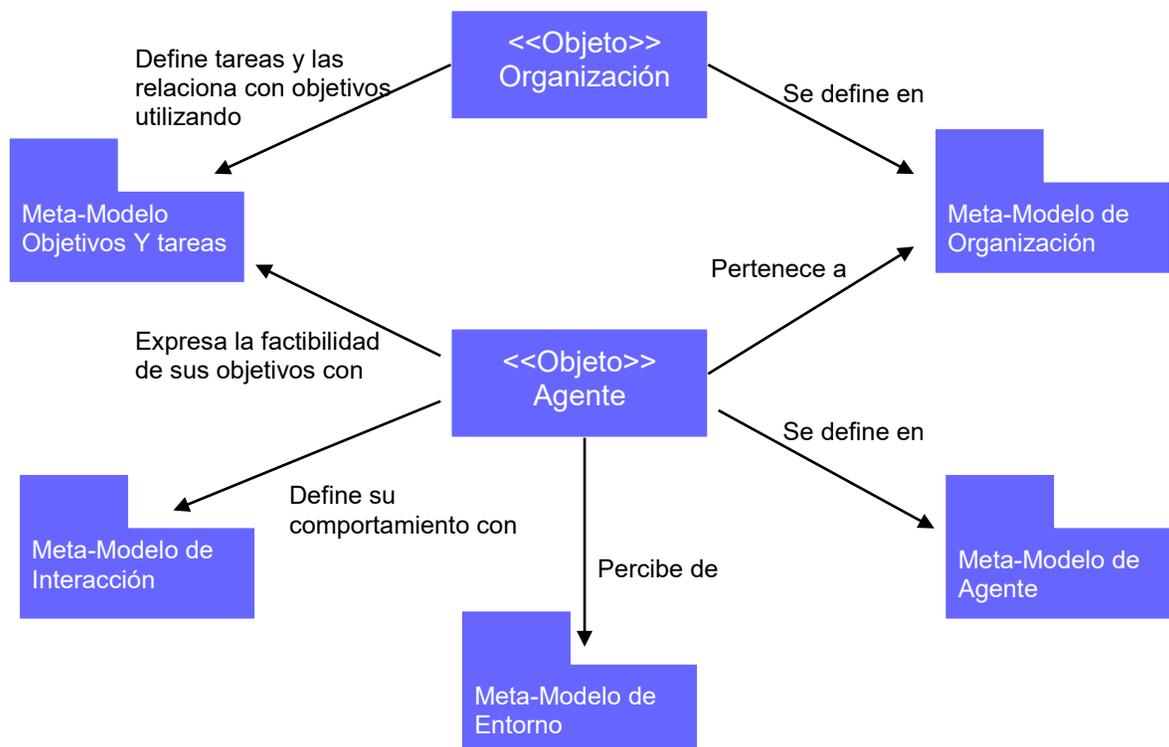
- Un lenguaje visual para la definición del SMA: este lenguaje ha sido creado para permitir el trabajo con el SMA de forma similar a como UML fue creado para trabajar con Sistemas orientados a objetos. Comprende una serie de jerarquía de conceptos y relaciones organizadas en vistas. Los conceptos, relaciones y vistas son definidas en meta-modelos.
- Integración con el ciclo de vida de desarrollo de software
- Desarrollo de herramientas: Provee una herramienta llamada INGENIAS IDE, que permite el modelado de las diferentes vistas del Sistema.

2.4.1 Definición del SMA con INGENIAS. INGENIAS hace una aproximación general para especificar el SMA que consiste en dividir el problema en aspectos mas concretos que forman las diferentes vistas del sistema. Cada vista esta descrita en un meta-modelo, el cual especifica una clase de gramática por medio de la cual se construirán los Meta-modelos. Un Meta-modelo describe qué debe tener un SMA por medio de los cuales se identifican entidades que pueden aparecer en el futuro SMA.

Los meta-modelos aplicados durante el análisis del proyecto actúan como guía, identificando las entidades a definir. En la fase de diseño, los meta-modelos son refinados, identificando nuevos componentes y relaciones entre ellos, para alcanzar un gran nivel de detalle.

Como se ve en la figura 6, INGENIAS incorpora nuevas herramientas de soporte y ejemplos de desarrollo⁴⁷, definiendo un conjunto de meta-modelos (una descripción de alto nivel de qué elementos tiene un modelo) con los que hay que describir el Sistema. Los Meta-modelos⁴⁸ indican qué hace falta para describir: Agentes aislados, organizaciones de Agentes, el entorno, interacciones entre Agentes o roles, tareas y objetivos. A continuación se presenta la relación existente entre los diferentes meta-modelos con las dos entidades principales que ofrece la Metodología INGENIAS.

Figura 6. Relaciones entre los diferentes Meta-modelos y las dos entidades principales, la organización y el agente



Fuente. Ferber, J. y Gutknecht, O.: A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems. Actas de conferencia. Proceedings of the

⁴⁷ WAGNER G. Agent-Oriented Analysis and Design of Organizational Information Systems. In *Proc. of Fourth IEEE International Baltic Workshop on Databases and Information Systems, Vilnius (Lithuania)*, May 2000

⁴⁸ Ferber, J. y Gutknecht, O.: A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems. Actas de conferencia. Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS98), IEEE CS Press. 1998

Third International Conference on Multi-Agent Systems (ICMAS98), IEEE CS Press. 1998

2.4.2 Meta-Modelo de Agente. Es usado para describir Agentes particulares excluyendo las interacciones con otros Agentes basándose en la funcionalidad del agente y en el diseño de su control. Proporciona información acerca de responsabilidades (son las tareas que sabe ejecutar y de los objetivos que se compromete a alcanzar) y el comportamiento (conjunto de llamadas a procedimiento UML, paso de mensajes entre Agentes MASE, MESSAGE o transiciones en máquinas de estado).

La justificación para dividir esta definición es que desde el punto de vista de construcción del agente, cada agente satisface las necesidades para las que fue diseñado. Por ello se hace referencia a la inteligencia o autonomía del agente.

La **inteligencia** surge al considerar mejoras en el control de ejecución de las tareas (planificación y parametrización de tareas, generación de nuevas tareas como composición de existentes) o en las tareas en sí mismas (tareas de comunicación que eligen colaboradores que se adecuen a las necesidades del momento o predicción del comportamiento del usuario). La inteligencia puede surgir del diseño del propio agente, incorporando algoritmos que permitan la toma de decisiones o como resultado de la combinación de algoritmos simples⁴⁹.

La **autonomía** viene soportada por el diseño y por la inteligencia del agente. Se consideran las diferentes situaciones en que se encuentre el agente, teniendo en cuenta también posibles cambios en el entorno (aparición de nuevos agentes, no disponibilidad de la red, agotamiento de recursos de la máquina). Interesa que el

⁴⁹ FERBER, J. y Gutknecht, O.: A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems. Actas de conferencia. Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS98), IEEE CS Press. 1998

Fuente. WOOLDRIDGE, Michael et al. 0-471-49691-X, Agent – oriented software engineering for Internet applications. An introduction to Multiagent Systems October 2002.

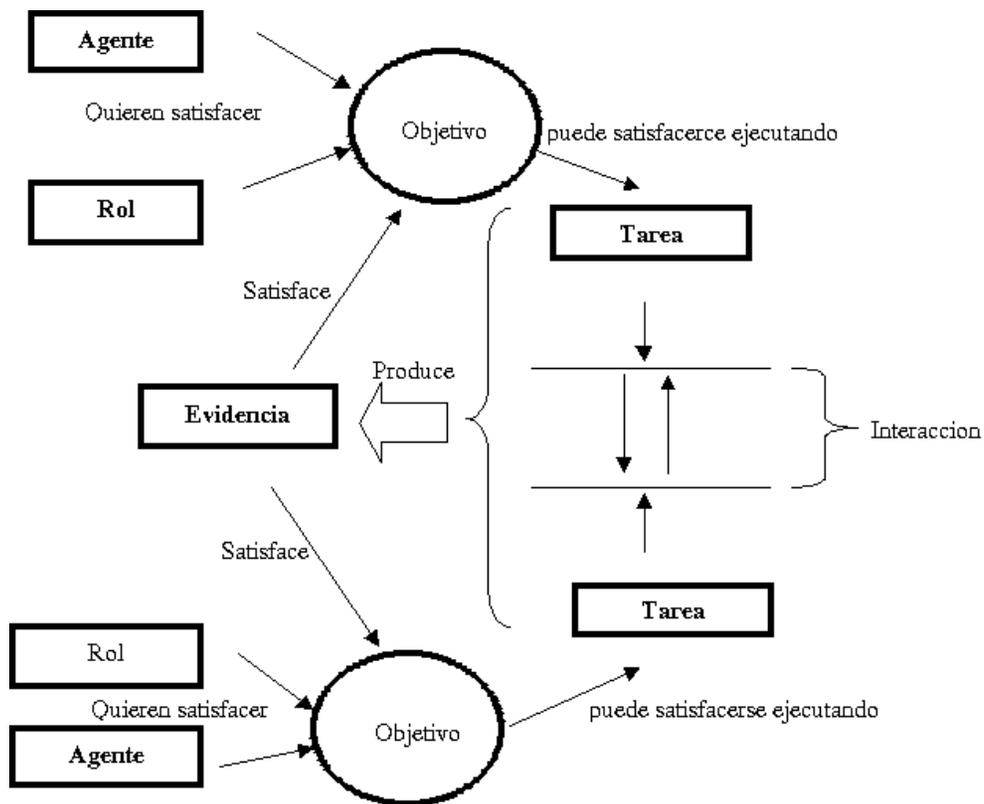
En el Meta-Modelo de Agente (figura 7), un agente es una entidad autónoma, la cual se caracteriza por tener propósitos y una única identidad. La entidad agente, tiene la capacidad de perseguir objetivos y, mediante las asociaciones con roles y tareas, de alcanzarlos de manera autónoma. Un agente, siguiendo la definición de Newell⁵⁰ es una entidad autónoma que actúa en el nivel de conocimiento y que se basa en el principio de racionalidad. El principio de racionalidad dice que la entidad realiza sólo aquellas acciones que le llevan a satisfacer sus objetivos.

2.4.2.1 Control del agente. El meta-modelo de Agente soporta un paradigma de control⁵¹ en el que el agente juega cierto rol, el cual le lleva a perseguir unos objetivos. Estos objetivos se alcanzan cuando el mundo, tal como lo concibe el agente, se ha modificado. Como se muestra en la figura 8, las tareas son las encargadas de producir esta modificación la cual se expresa mediante la producción de evidencias que pueden ser nuevas entidades mentales o eventos disparados por el entorno. Este esquema se incluye en el Meta-modelo a través del estado mental, el gestor de estado mental y el procesador de estado mental.

⁵⁰ WOOLDRIDGE and N. R. Jennings. Agent theories, architectures, and languages: A survey. In M. Wooldridge and N. R. Jennings, editors, Intelligent Agents: Theories, Architectures, and Languages (LNAI Volume 890), Springer-Verlag: Heidelberg, Germany, Jan. 1995

⁵¹ FERBER, J. y Gutknecht, O.: A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems. Actas de conferencia. Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS98), IEEE CS Press. 1998.

Figura 8. Conceptos relevantes en el control del agente



Fuente. FERBER, J. y Gutknecht, O.: A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems. Actas de conferencia. Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS98), IEEE CS Press. 1998

El *estado mental* es toda aquella información que permite al agente tomar decisiones. Esta información es gestionada y procesada para producir las decisiones del agente a partir de unos datos.

El Gestor Del Estado Mental tiene como propósito desarrollar la evolución del estado mental mediante las operaciones de creación, destrucción, modificación y monitorización del conocimiento del agente. Es responsable de mantener la coherencia del conocimiento almacenado y de hacerlo evolucionar.

El *Procesador del Estado Mental* se propone la toma de decisiones en sí, el control del agente. Las decisiones se pueden producir de forma algorítmica, en función de las tareas a ejecutar y el beneficio a obtener, u obedecer a la consecución de objetivos eligiendo secuencia de tareas a ejecutar.

2.4.3 Meta-Modelo de Objetivos y Tareas. El meta-modelo de objetivos y tareas tiene como propósito recoger las motivaciones del SMA, definir las acciones identificadas en los modelos de organización, interacciones o de agentes y cómo afectan estas acciones a sus responsables. El meta-modelo objetivos y tareas expresa cuáles son las consecuencias de ejecutar las tareas y por qué se deberían llegar a ejecutar.

Este meta-modelo se usa para expresar la motivación que hay detrás de las tareas y qué opciones de actuación se le presentan a un agente en un momento dado.

2.4.3.1 Tareas en los Sistemas Multi-Agente. El concepto *tarea* se puede ver como transformaciones del estado global siendo útil porque concibe la tarea como pre-condiciones y post-condiciones (el cómo son estas pre y post-condiciones se

discutirá más tarde) permitiendo su incorporación en mecanismos de planificación y razonamiento, o tarea como proceso siendo un conjunto de instrucciones que han de ejecutarse.

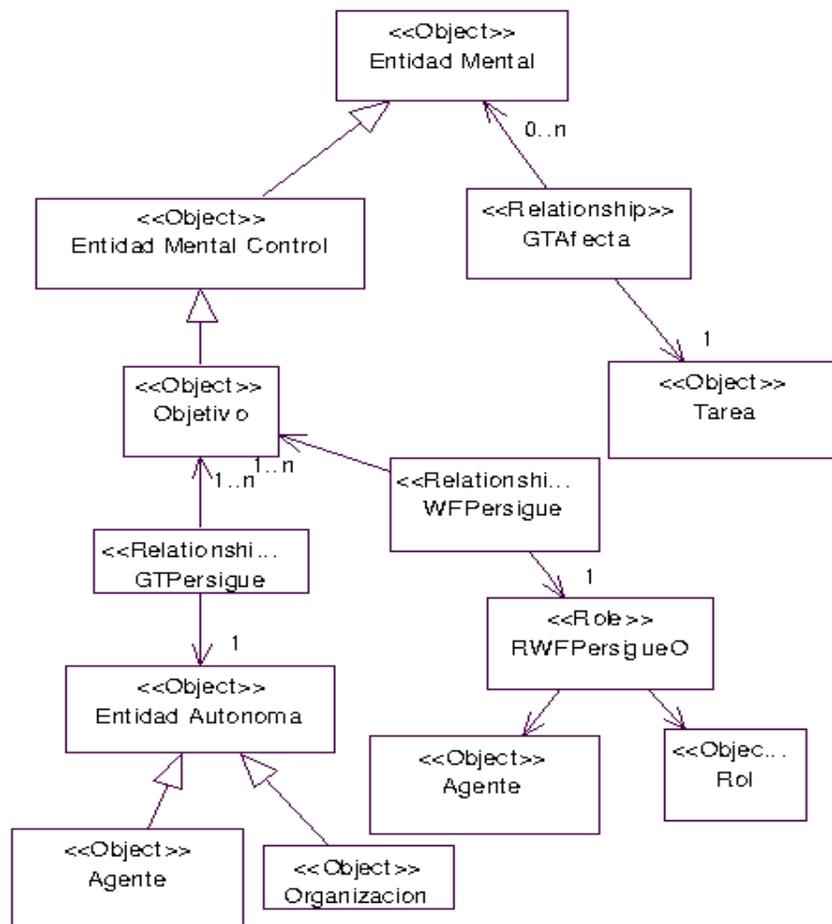
2.4.3.2 Objetivos en los Sistemas Multi-Agente. Los objetivos se emplean para razonar acerca de las posibles alternativas que se le presentan a un agente en un momento determinado. Para representar estas alternativas, existen dos tendencias:

- Objetivos como agregación. El objetivo se ve como una descripción del estado del mundo a alcanzar. Cuando la descripción del estado del mundo se hace con un conjunto de predicados, el objetivo se convierte en una agregación de estos predicados.
- Objetivos como entes. Los objetivos no son agregaciones de predicados, como antes, sino de entes autorepresentativos. Un objetivo, según ⁵², es una unidad mental resultado de impulsos (del propio agente) y solicitudes, de otros agentes.

2.4.3.3 Presentación del meta-modelo de tareas y objetivos. En el meta-modelo se refleja la relación de los objetivos con los agentes, roles y organizaciones. Una organización, como *entidad autónoma* al igual que un agente persigue objetivos (*GTPersigue*). Los roles, por el contrario, se asocian con los objetivos en el marco de los flujos de trabajo como se observa en la figura 9. Los agentes, por poder aparecer en los flujos de trabajo y ser una *entidad autónoma*, como la organización, pueden asociarse con cualquiera de las dos.

⁵² FERBER, J. y Gutknecht, O.: A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems. Actas de conferencia. Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS98), IEEE CS Press. 1998.

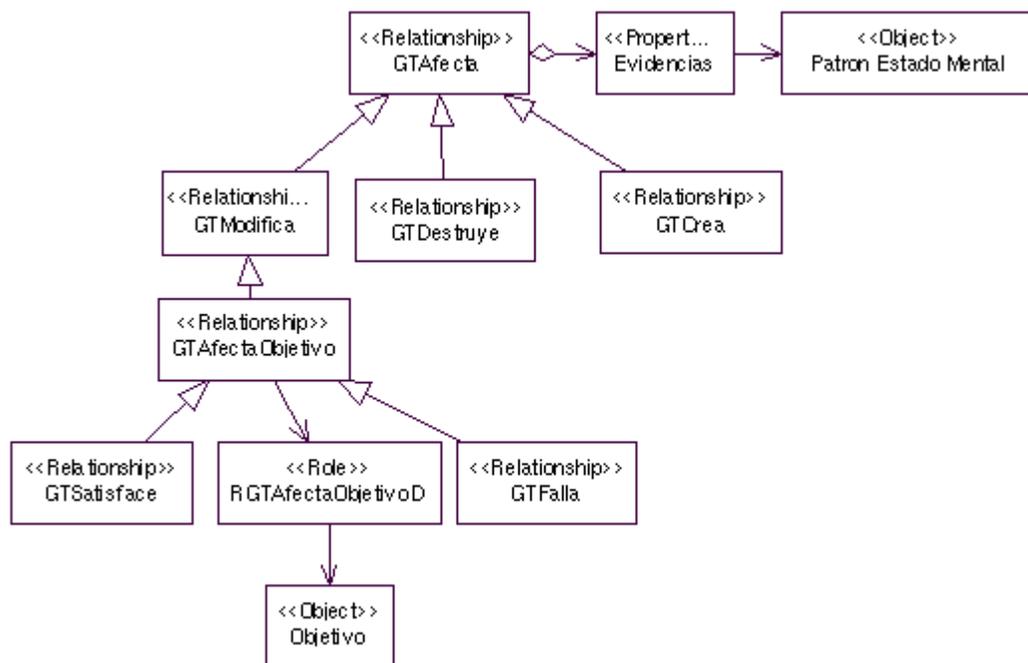
Figura 9. Relaciones entre objetivos y tareas



Fuente. RICH, E. y Knight, K.: *Artificial Intelligence*. Libro completo. McGraw-Hill. 1990

Las tareas se asocian con los objetivos mediante la meta-relación GTAfecta como se muestra en la figura 9, cuya semántica es que la ejecución de la tarea afecta de una forma específica a una entidad mental. El ámbito de actuación de la tarea se limita a las entidades mentales del ejecutor de la tarea. La modificación del estado mental de otros agentes se modela haciendo que una tarea produzca una interacción con otro agente. Esta interacción provoca la ejecución en el otro agente de tareas que son las responsables finales de realizar el cambio.

Figura 10. Formas en que una tarea afecta a un objetivo



Fuente. RICH, E. y Knight, K.: *Artificial Intelligence*. Libro completo. McGraw-Hill. 1990

En la figura 10, se distinguen tres formas de alterar el estado mental: destruyendo una o varias entidades (GTDestruye), creándolas (GTCrea) o modificándolas (GTModifica). Todas ellas se decoran con un patrón de estado mental para indicar en qué condiciones la creación, modificación o destrucción tiene lugar. Para el

caso de la modificación y destrucción, se puede citar dentro de estos patrones una entidad del mismo tipo que el que se va a ver afectado para reflejar dependencias de las entidades a modificar o destruir.

Para el caso de los objetivos, se ha especializado la relación GTModifica. Se dispone de dos especializaciones que concretan la forma de modificar un objetivo: satisfaciéndolo (GTSatisface) o haciéndolo fallar (GTFalla). Estas dos últimas se condicionan a la existencia de ciertas evidencias que indiquen cuándo tiene lugar el fallo o la satisfacción del objetivo. Cuando se alcanza un objetivo o cuando este fracasa se consideran dos enfoques:

- Delegar en el gestor y procesador de estado mental. De forma genérica se establece cual es el tratamiento de los objetivos fracasados o satisfechos. Este tratamiento entraría dentro de las especificaciones del gestor y procesador de estado mental.
- Definir nuevas tareas para tratar con objetivos fracasados o satisfechos. El cómo se realiza la gestión de entidades se detalla mediante tareas que destruyen, crean o modifican las entidades mentales existentes.

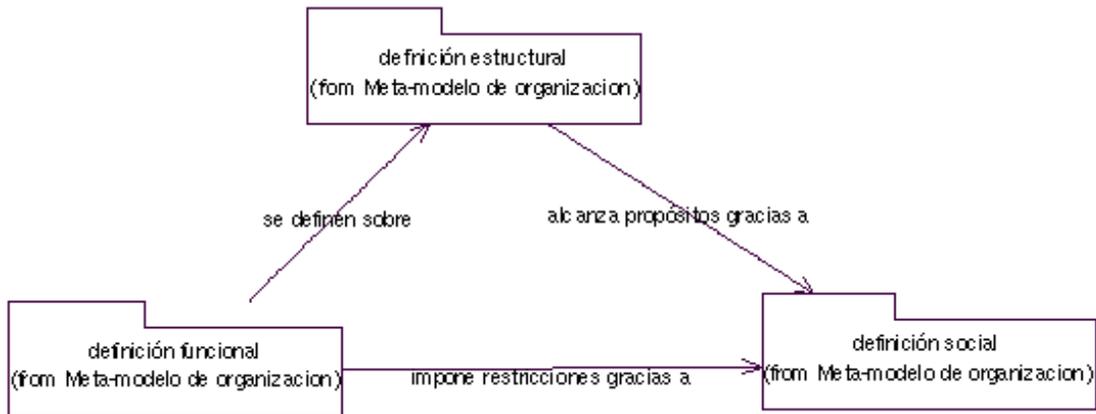
2.4.4 Meta-Modelo de Organización. El meta-modelo de Organización es la arquitectura del Sistema en un SMA. El valor principal son los flujos de trabajo que define. Del estudio de estos flujos surgen nuevas interacciones que reflejan con detalle cómo se coordinan los participantes del flujo. También define restricciones en el comportamiento de los Agentes mediante relaciones como la de subordinación.

Es la arquitectura del Sistema en un SMA, se encarga de definir flujos de trabajo y describe cuales componentes del Sistema (agentes, roles, recursos y aplicaciones) son agrupados, cuales tareas son ejecutadas en común, cuales objetivos son compartidos y que restricciones existen en la interacción entre

agentes. Esas restricciones son expresadas en forma de relaciones de subordinación y cliente servidor.

Para simplificar el meta-modelo se han separado tres aspectos del mismo, como se muestra en la siguiente figura 11.

Figura 11. Vistas de una organización

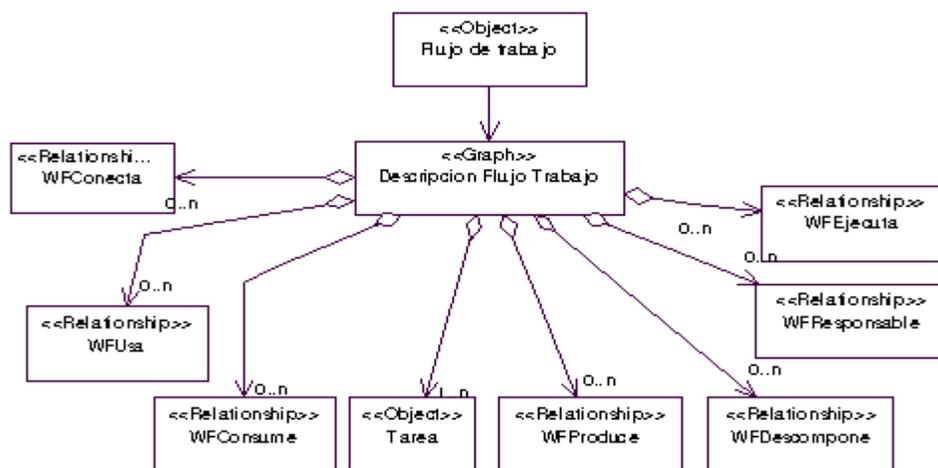


Fuente. WHITAKER, R.:Self-Organization, Autopoiesis, and Enterprises. ACM SIGGROUP. Sitio WEB: <http://www.acm.org/siggroup/auto/Main.html>

La figura 11 muestra las tres vistas de una organización y la estructura donde define los elementos principales que componen la organización y cómo se construye ésta a partir de ellos. Las relaciones sociales establecen relaciones de alto nivel entre los componentes para poner restricciones al comportamiento de la

organización. Por último, la definición funcional establece qué ofrece la organización y cómo se lleva ésta a cabo⁵³.

Figura 12. Meta-modelo de organización. Elementos que componen un flujo de trabajo



Fuente. Workflow Management Coalition: The Workflow Management Coalition Specification: Workflow Management Coalition Terminology & Glossary. Informe. 1999

En la figura 12 se muestran los elementos que componen un flujo de trabajo donde algunas de sus relaciones (concretamente WFConsume, WFUsa, WFProduce, WFDescompone) se explicaran mas adelante en el meta-modelo de objetivos y

⁵³ MATURANA, H.: Autopoiesis, Structural Coupling and Cognition. International Society for the System(s) Science(s) (ISSS). [online] Available from World Wide Web: <http://www.issss.org/maturana.htm> [cited 1 march 2004]

tareas. WFEspecificaEjecucion indica que la especificación concreta de las condiciones de ejecución de una tarea se presenta dentro de una interacción. Se ha adoptado esta solución para correlacionar tareas ejecutadas por distintos actores. WFConecta se corresponde con la conexión de las salidas (instancias de WFProduce) de una tarea con las entradas de otra (instancias de WFConecta). La conexión no es completa en el sentido de que hay necesidad de que todo lo producido por una tarea deba ser consumido por la otra. Para determinar exactamente qué se consume, hay que considerar las instancias de WFConsume.

Se permite que un flujo de trabajo se descomponga en otros flujos de trabajo o tareas. De esta forma se pueden generar definiciones incrementales de las tareas que componen el flujo de trabajo. No se permite la descomposición de una tarea en flujos de trabajo.

Las tareas son ejecutadas por agentes que son directamente responsables de ellas (mediante instancias de WFResponsable) o indirectamente a través de roles (mediante instancias de WFJuega), como indica la figura 12. Que un agente o rol sea responsable de la ejecución de una tarea, significa que el agente sabe ejecutar esa tarea.

Entre las distintas entidades de la organización surgen dieciséis posibles combinaciones de dependencias entre organizaciones, grupos, roles y agentes.

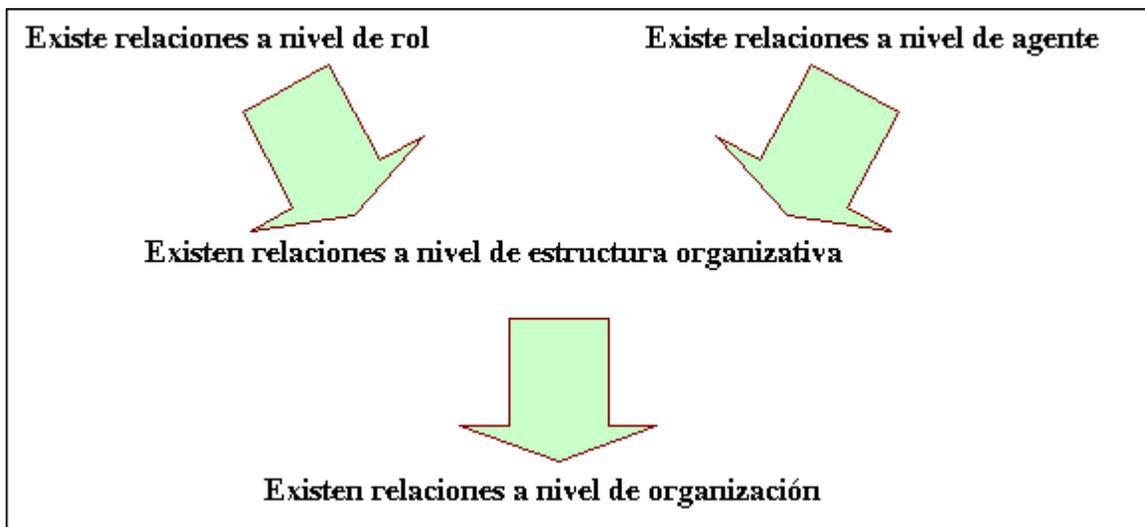
Para limitar y estructurar estas relaciones, se admiten sólo relaciones entre entidades del mismo tipo como se diagrama en la figura 13. Sobre las relaciones resultantes se plantea una política de aplicación. Se establece que entre dos entidades de la organización pueden existir como máximo relaciones a tres niveles. De mayor a menor nivel, estos niveles son:

Nivel organizativo. Se trata de las relaciones organizacionales. Se dan entre las entidades de más alto nivel y son el equivalente a las relaciones interempresariales.

Nivel de estructura de organización. Se trata de las relaciones estructurales. Serían las equivalentes a relaciones entre departamentos.

Nivel de agente y rol. Se trata de las relaciones de agencia. Serían equivalentes a relaciones directas entre el personal.

Figura 13. Relaciones sociales



Fuente. SYCARA, K., Klusch, M., idof, S., and Lu, J., Dynamic Service Matchmaking among Agents in Open Information Environments, Journal ACM SIGMOD Record , Special Issue on Semantic Interoperability in Global Information Systems, 1999

No puede haber relaciones sociales entre dos entidades entre las cuales no exista una relación a nivel inmediatamente superior.

2.4.5 Meta-Modelo de Interacción. **ve en la figura**

El rol de las interacciones en la ISOA es fundamental ya que las interacciones identifican dependencias entre los componentes y contribuyen a la especificación del comportamiento de los componentes así como la funcionalidad asociada.

2.4.5.1 Análisis y diseño de interacciones. Las interacciones determinan el comportamiento de los agentes mostrando cuál es su reacción cuando actúan sobre ellos. Y cómo el comportamiento va a ser función de los objetivos de los agentes y las tareas a ejecutar, se puede concluir que existe un importante vínculo entre interacciones, objetivos y tareas.

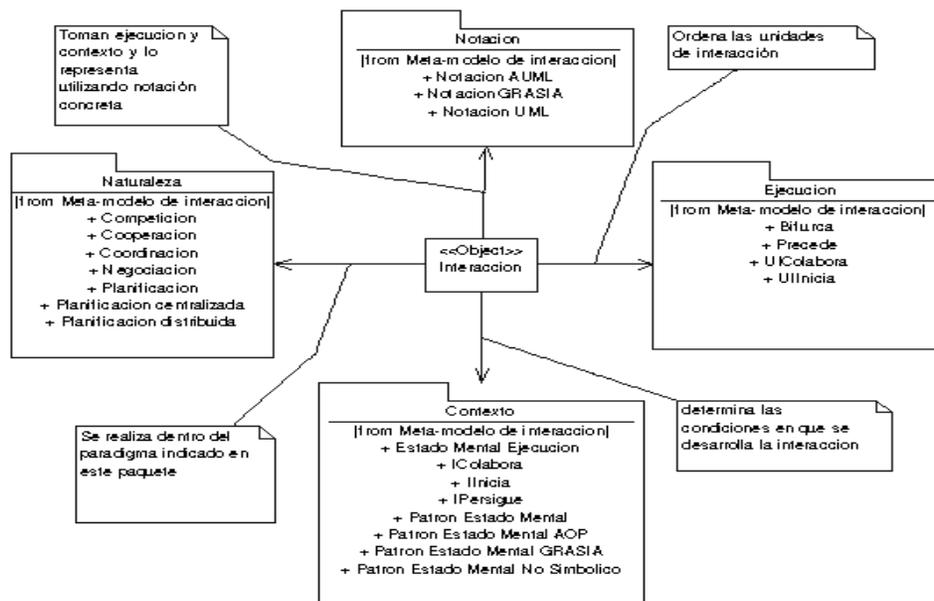
El nivel de abstracción en el que se definen las interacciones cambia del análisis al diseño. En el diseño, la especificación completa de una interacción tiene que cubrir:

- Los actores que participan. Un actor debería mostrar por qué está participando en la interacción.
- La definición de unidades de interacción. La naturaleza de la unidad de interacción determina cómo tiene que procesarla el receptor. Una unidad de interacción puede ser tan simple como un paso de mensaje o mas compleja aún nombrada como mensajes activos que se caracterizan por hacerse responsables de la interacción con el receptor de una forma personal dependiendo de las capacidades del receptor.

- Un orden sobre las unidades de interacción. Las unidades de interacción se organizan siguiendo un protocolo estándar o adecuado a una situación concreta.
- Acciones ejecutadas en la interacción. Incluye detalles sobre: Criterios para decidir cuándo ejecutar una tarea y Consecuencias de la ejecución de una tarea.
- Definición del contexto de la interacción. El contexto consiste en detallar qué ocurre en el Sistema cuando se inicia la interacción, mientras se desarrolla y concluye. Esta información se proporciona indicando qué actores participan, qué motivos los impulsan a hacerlo, qué objetivos persigue la interacción y qué estados mentales se requieren de los agentes durante la ejecución de la interacción.
- Un modelo de control. El control asegura que la interacción va a desarrollarse según fue definida. Este control debería tener en cuenta que en la mayoría de los casos se permite llevar varias interacciones en paralelo.

2.4.5.2 Elementos de la interacción. El meta-modelo de interacción se construye sobre agentes, roles, objetivos, interacciones y unidades de interacción. Los agentes y roles son los actores de las interacciones. En las interacciones se ejecutan unidades de interacción (pasos de mensaje, lectura y escritura en un espacio de tuplas) en las que hay un iniciador (emisor) y colaboradores (receptores). Además, se justifica la participación de los actores en la interacción y la existencia de la interacción en sí mediante objetivos.

Figura 14. Relación entre los diferentes aspectos de la interacción



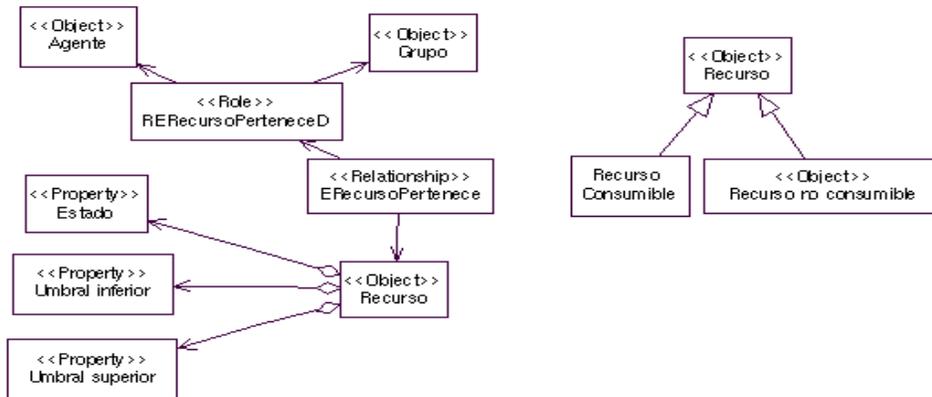
Fuente. MALONE, T. W. and Crowston, K., *The Interdisciplinary Study of Coordination*, ACM Computing Survey, vol. 26, no. 1, pp. 87-119, Mar.1994

La relación de aspectos es una Ayuda para situar la interacción en el marco del SMA como muestra la figura 14, estableciendo en qué condiciones se iniciarán las interacciones. Esta información se utiliza para codificar el comportamiento del agente en forma de reglas de producción o para generar métodos de validación de la interacción, como métodos de detección de interbloqueos. En el meta-modelo, el contexto se expresa con los estados mentales en ejecución que deben reunir los agentes y con las meta-relaciones IPersigue, IColabora y IInicia.

2.4.6 Meta-Modelo de Entorno. El meta-modelo de entorno comienza restringiendo el tipo de elementos que van a aparecer como se observa en la figura 15. Se distinguen tres posibles tipos: agentes, recursos y aplicaciones. Por recurso se entiende todo aquel objeto del entorno que no proporciona una funcionalidad concreta, pero que es indispensable para la ejecución de tareas y cuyo uso se restringe a consumir o restituir. Cuando el uso sea más complejo,

como la funcionalidad requerida de una base de datos, se empleará el término aplicación.

Figura 15. Meta-modelo de entorno



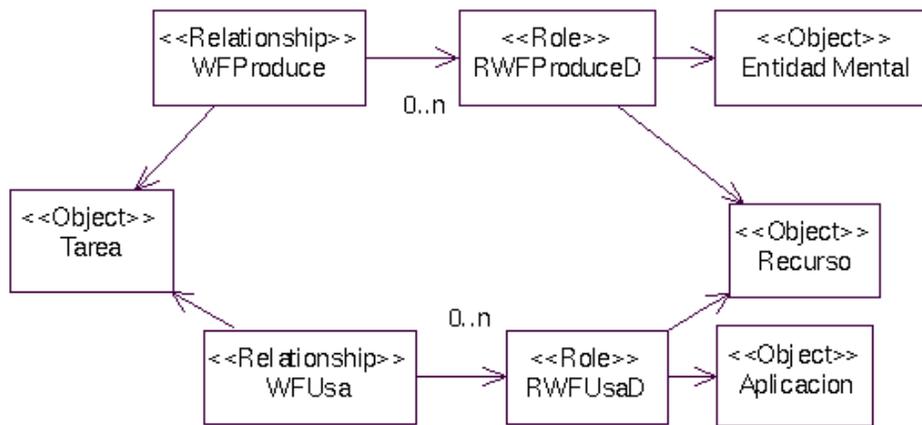
Fuente. Decker, S. Keith: *Environment Centered Analysis and Design of Coordination Mechanisms*. Informe. Department of Computer Science, University of Massachusetts. 1995

Los recursos pertenecen a un agente o a un grupo (RERecursoPertenece). Esta relación es similar a (OContieneGrupo) donde se indica únicamente si el grupo tiene un recurso. Dentro de esta relación se permite también que el recurso esté bajo el cuidado de un agente. Cada recurso se caracteriza con tres propiedades básicas: el estado actual, el umbral inferior y el umbral superior. Se ha incorporado el tipo de recurso (consumible o no consumible) mediante herencia.

Las aplicaciones se caracterizan por poseer un conjunto de operaciones con una signatura convencional como se observa en la figura 16. Las precondiciones y poscondiciones se trasladan directamente de las especificaciones de la aplicación (las aplicaciones ya están desarrolladas así como la interfaz para interactuar con ellas) cuando esta sea un software existente. Estas operaciones se utilizan para modelar la percepción del agente. Inicialmente se distinguen sólo dos tipos: percepción: por muestreo y por notificación. En ambos tipos se asocia la

percepción (*EPercibe*) con una operación concreta. En el caso de muestreo, *operación* es una operación que se va a ejecutar con una frecuencia determinada. Las aplicaciones pueden existir con anterioridad al desarrollo actual (*AplicacionEntorno*) o ser desarrolladas *ad-hoc* para los propósitos actuales (*AplicacionInterna*). Las primeras se obtienen de la captura de requisitos, mientras que las segundas se generan mediante técnicas convencionales de ingeniería del software.

Figura 16. Meta-modelo de entorno. Tareas, recursos y aplicaciones



Fuente. RUSSELL, S. y Norvig, P: Artificial Intelligence: a modern approach. Libro completo. Prentice Hall. 1995

Tareas, recursos y aplicaciones se relacionan. Como ya se ha visto en el meta-modelo de tareas y objetivos y en el meta-modelo de organización, la tarea consume y restituye recursos, pero también usa aplicaciones

3 CUADERNOS ELECTRONICOS

Tras grandes dificultades que se presentan en la toma y búsqueda de anotaciones, surge el concepto de Cuadernos Electrónicos⁵⁴ que es el equivalente a un Cuaderno de investigación, el cual está diseñado para guardar información de tipo texto, ecuaciones, imágenes, graficas, notas personales u otro tipo de datos mediante el computador o directamente desde instrumentos

⁵⁴ GEIST, AI Design of The DOE2000 Electronic Notebook : The Electronic Notebook Architecture. Berkeley California. December 1997

tecnológicos permitiendo ingresar información desde distintas ubicaciones geográficas, para hacer mas eficiente y ofrecer una mayor colaboración en el trabajo de equipo. El Software de Cuadernos Electrónicos varía de acuerdo a su apariencia y funcionalidad cumpliendo con las funciones básicas de un Cuaderno de papel.

Un cuaderno electrónico es el equivalente electrónico de un cuaderno de papel de investigación, donde en lugar de guardar la información sobre papel, bosquejos, texto, ecuaciones, gráficos, firmas y otros datos que se registran (los cuales se puede leer y navegar como en un cuaderno de papel) se hace a través de una computadora, implicando la entrada del teclado, mouse, archivos de la imagen, micrófono o directamente de los instrumentos científicos, permitiendo una entrada más fácil de datos científicos y de la capacidad para los colaboradores en diversas localizaciones geográficas de compartir el expediente de ideas, de datos y de acontecimientos de los experimentos y de los programas de investigación comunes.

Generalmente, el contenido de los cuadernos electrónicos se mantiene privado a menos que sea necesario para probar cuando una idea fue registrada o se quiera trabajar en colaboración con otros investigadores en determinadas areas o proyectos de investigación y que utilizan el instrumento y el servicio como documentación del estado del instrumento y los utilizan.

Existen métodos para guardar información en un Cuaderno en las diferentes disciplinas científicas tales como colocar fecha a todas las nuevas anotaciones, escribir texto en páginas consecutivas, hacer bosquejos rápidos, dibujar tablas, agregar anexos en hojas separadas (figuras, fotos, graficas y resultados de computadora).

El cuaderno electrónico está siendo reconocido una como herramienta importante que necesita ser desarrollada para que los científicos y los ingenieros hagan la experimentación y trabajen en colaboración desde distintos puntos geográficos, convirtiéndose en un medio por el cual los investigadores pueden registrar remotamente aspectos de los experimentos que se conducen con gran facilidad.

Pero el uso de cuadernos electrónicos va más allá de documentar el uso de instrumentos alejados. Pueden ser utilizados como los cuadernos privados que documentan la información e ideas personales, o solo como cuaderno del "proyecto" compartido por un grupo de colaboradores como los medios de compartir ideas científicas entre sí mismos.

Los Cuadernos Electrónicos se caracterizan porque se puede compartir en grupo de investigadores, ingresar remotamente, no será extraviado, olvidado o accidentalmente destruido (uso de backup), es fácil de incorporar a archivos de computadoras, busca fácilmente la información, incluye archivos de multimedia (audio/video) y contiene vínculos con otra información.

Un Cuaderno Electrónico debe tener en cuenta aspectos como:

- Fácil uso: se debe construir una interfaz donde la entrada de información sea muy intuitiva y el investigador deduzca el funcionamiento del Cuaderno Electrónico sin haber leído un manual. También debe proveer métodos rápidos y simples de ingresar texto, resultados y usar métodos comunes de computadora como copiar/pegar, Arrastrar/soltar e insertar archivo. La primera llave a la aceptación de empleo fácil y amplia del cuaderno electrónico es no desviarse de la funcionalidad del cuaderno de papel básico. La meta inicial es reproducir su hábito en un formato electrónico. El hábito para un cuaderno varía de un solo investigador que apunta pensamientos personales a un grupo de ajustes y de resultados del instrumento de grabación de los científicos. No tiene sentido de cargar el

diseño básico del cuaderno con las características que pueden nunca ser utilizadas y servicio justo para complicar la interfaz usada.

La segunda llave a la facilidad de empleo es crear una interfaz que proporcione la entrada intuitiva de la información. Un investigador debe poder calcular fuera de cómo utilizar el cuaderno electrónico sin tener que leer un manual del usuario. El cuaderno electrónico debe proporcionar los métodos rápidos y simples para ingresar el texto y bosquejos apenas como los cuadernos de papel. Debe también proporcionar los métodos comunes de computadora entrados como cortar/pegar, arrastrar/soltar y archivo de relleno.

La información que necesita ser entrada en un cuaderno varía de proyecto a proyecto y de investigador a investigador. El deseo de tener cualquier tipo de información fácil entrar requiere que la interfaz sea extensible. El usuario del cuaderno o los terceros podría extender el tipo de objetos apoyados para modificar un cuaderno para requisitos particulares.

- Seguridad: El acceso a un Cuaderno Electrónico es fundamental, en consecuencia debe certificar que las entradas de un usuario sean correctas mediante una lista de usuarios autorizados manejando un alto nivel de privacidad para el grupo de Investigación. Por otra parte, un Cuaderno Electrónico puede ser usado por una o mas personas en colaboración remota ofreciendo privilegios para documentar ciertos experimentos, además de insertar fechas y almacenar digitalmente cada entrada y añadirla a la ultima pagina del Cuaderno Electrónico. Para tener acceso a un cuaderno electrónico particular, su contenido y la autenticación de entradas en un cuaderno se debe trabajar las ediciones de seguridad fundamentales que cualquier cuaderno debe tener. Cada cuaderno tiene una lista de usuarios autorizados, así el aislamiento se maneja en el nivel de acceso del cuaderno. Si es una lista de una clase de permiso, entonces

es un cuaderno privado. Si la lista es mayor de una clase de permiso entonces es un cuaderno de grupo. Un cuaderno de grupo se puede instalar para la colaboración alejada o podría ser utilizado por la lista de usuarios autorizados de un instrumento científico particular para documentar experimentos. Para proporcionar la seguridad de entradas, el diseño del cuaderno debe fechar automáticamente y digitar la firma de cada entrada, después añade la entrada a la página pasada del cuaderno. Una página del cuaderno se puede certificar ante notario por medio de un botón.

La notarización se debe hacer electrónicamente de una manera que el notario nunca ve la información en la página (que sería de otra manera un agujero potencial del espionaje). Regularmente, una copia (de papel) física de un picadillo de las páginas certificadas ante notario de todo el cuaderno es certificado ante notario y guardado por la gerencia de expedientes para proporcionar un segundo nivel de la prueba con el cual las entradas no se han tratado de forzar. Una sola hoja del papel podía contener los derechos de autor de todos los cuadernos electrónicos activos a través de una organización.

- Interoperabilidad y expansibilidad: los usuarios acceden y actualizan el Cuaderno Electrónico desde diferentes tipos de computadores con distintos Sistemas operativos. Los científicos serán quienes tendrán acceso y podrán actualizar los cuadernos electrónicos de diversos tipos de computadoras. El software del Cuaderno Electrónico se debe diseñar para funcionar a través de plataformas de la PC, del Mac y de Unix. Es también deseable de un punto de vista de la tecnología de dotación lógica crear un paquete de software que sea portable a una amplia gama de las plataformas que cree varios paquetes cada uno apuntado a las plataformas individuales.

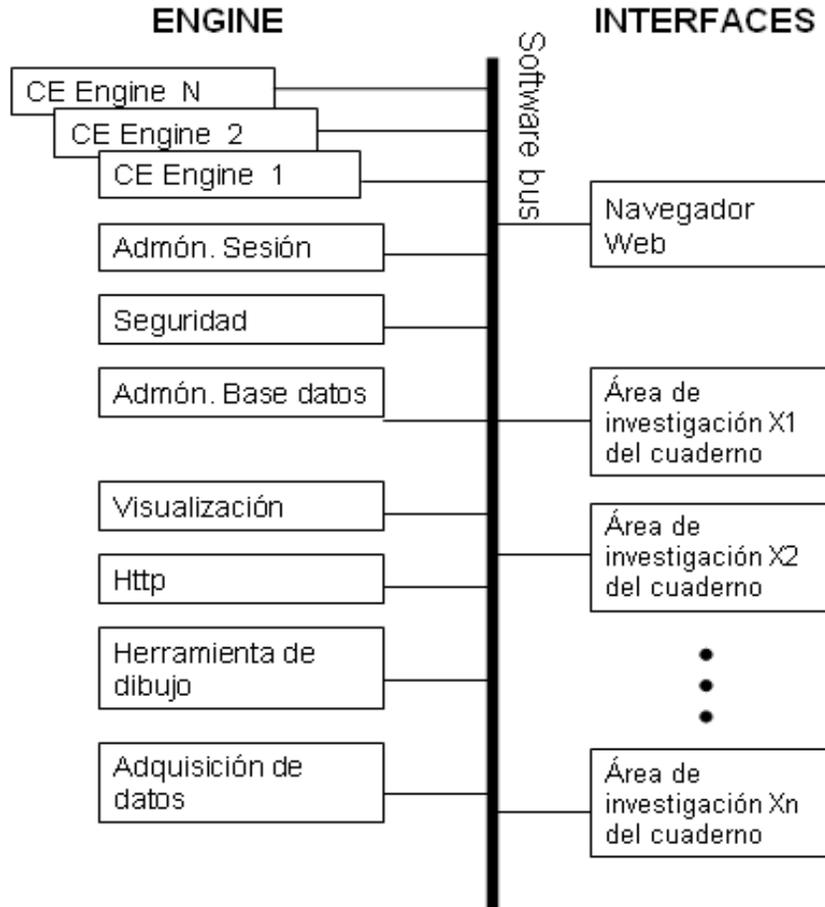
Porque las necesidades de diversos proyectos y de científicos varían, el diseño del cuaderno debe proporcionar interfaces generales para la extensión de la funcionalidad del cuaderno ambos por agregaciones de los terceros y el arreglo para requisitos particulares del usuario.

Lo ideal es crear un cuaderno electrónico extensible, interoperable que los científicos y los ingenieros encuentren más útil que sus cuadernos del documento.

Hoy en día, existen proyectos similares de Cuadernos Electrónicos como el prototipo ORNL⁵⁵ que ha sido distribuido y usado para proyectos en diversos países. A partir de este prototipo, ha surgido gran cantidad de proyectos de Cuadernos Electrónicos en distintas organizaciones, Instituciones educativas, en el campo de la medicina, grupos de investigación y para uso personal. Basados en la arquitectura del prototipo ORNL se desarrollará el Cuaderno Electrónico teniendo en cuenta algunos aspectos importantes como la toma de anotaciones, seguridad, administración de sesión e interfaces.

Figura 17. Arquitectura propuesta para cuadernos electrónicos

⁵⁵ GEIST, AI Design of The DOE2000 Electronic Notebook : The Electronic Notebook Architecture. Berkeley California. December 1997



Fuente. GEIST, AI Design of The DOE2000 Electronic Notebook : The Electronic Notebook Architecture. Berkeley California. December 1997

Como se puede observar en la figura 17, los Cuadernos Electrónicos están desarrollados basados en la Web semántica⁵⁶ siendo ésta un área pujante en la unión de la Inteligencia Artificial y las tecnologías Web, que propone nuevas técnicas y paradigmas para la representación del conocimiento facilitando la acción de localizar, compartir e integrar recursos a través de la WWW⁵⁷.

⁵⁶ TIMBERNERS-LEE, J. Hendler, O Lassila. Fascinating facts about Tim Berners-Lee inventor of the World Wide Web. The Semantic Web 12-589-6587-AK25, <http://www.ideafinder.com/history/inventors/berners-lee.htm> Scientific American, May 2001

⁵⁷ RUSSELL, S. y Norvig, P: Artificial Intelligence: a modern approach. Libro completo. Prentice Hall. 1995.

Durante los últimos años, mediante el uso de la Web semántica se han venido desarrollando diversos Lenguajes y estándares para la definición de ontologías, entre ellos XML⁵⁸, RDF⁵⁹, DAML+OIL⁶⁰, y OWL⁶¹, respaldados por el consorcio W3C, uno de los principales promotores de la Web semántica.

Los cuadernos de papel son difundidos en la comunidad científica. Los investigadores guardan los cuadernos personales para registrar sus ideas, reuniones, y experimentos. El contenido de estos cuadernos generalmente se mantiene privado a menos que sea necesario para probar cuando una idea primero fue registrada. Los cuadernos se guardan en todos los instrumentos científicos importantes también. Estos cuadernos son compartidos por todos los investigadores que utilizan el instrumento y el servicio como documentación del estado del instrumento y los utilizan.

Las páginas o las entradas individuales del cuaderno nunca se ocultan de un investigador con el acceso válido al cuaderno. Ocultar datos podía ser peligroso y no fomenta una sensación de la colaboración. La información privada se debe mantener en el cuaderno personal de un investigador. Este es un procedimiento simple para transferir la información a partir de un cuaderno electrónico a otro si está deseado.

⁵⁸ BRAY, J. Paoli, C. M. Sperberg-McQueen, Nore Markup Language (XML) 1.0 (Second Edition). W3C Recommendation 6 October 2000. Disponible en: <http://www.w3.org/TR/REC-xml.html>

⁵⁹ JENNINGS, N.R. International Journal of Cooperative Information Systems, World Scientific Publishing Co., 0218-8430, Editado por M. P. Papazoglou, "Agent-based business process management.", N. R. Jennings, 1996, 105-130

⁶⁰ CONNOLLY, F. Van Harmelen, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. DAML+OIL Reference Description. W3C Note 18 December 2001. Disponible en <http://www.w3.org/TR/daml+oil-reference>.

⁶¹ DEAN, D. Connolly, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. OWL Web Ontology Language 1.0 Reference W3C Working Draft 29 July 2002. Disponible en <http://www.w3.org/TR/owl-ref>.

4. FASE DE ANALISIS Y DISEÑO

Para la realización de este proyecto se tuvo en cuenta toda la información relacionada acerca del proyecto de Cuadernos Electrónicos. El estudio de esta información se llevó a cabo con la búsqueda de artículos, proyectos de investigación, tesis realizadas con anterioridad y tutoriales los cuales se encuentran referenciados al final del documento.

En la fase de Análisis y diseño se tuvo en cuenta la Metodología INGENIAS para el desarrollo de Sistemas Multi-agentes. Gracias a la herramienta INGENIAS IDE se logro realizar el Modelado de los diagramas planteados por dicha Metodología de una forma más ordenada y especializada de acuerdo a la concepción de los Meta-modelos propuestos.

En el avance del proyecto APLICACIÓN DE LA METODOLOGÍA DE DESARROLLO INGENIAS Y TÉCNICAS DE WEB SEMÁNTICA EN LA IMPLEMENTACIÓN DE UN CUADERNO ELECTRÓNICO DE INVESTIGACIONES se tuvo en cuenta las herramientas de desarrollo necesarios para la concepción del sistema, tales como XML, JDOM, Java y Servidores Web (apache y Tomcat).

4.1 DESCRIPCION DEL SISTEMA

La metodología propuesta por INGENIAS se inicia con la etapa de concepción, la cual permite principalmente determinar el problema que se va a tratar, y crear un primer esquema del sistema. Para la realización de esta tarea, se seleccionó el análisis centrado en el usuario y en las anotaciones de un cuaderno electrónico, que se enfoca en el proceso de captura de datos, de manera que se puedan

comprender mejor todos sus requisitos y satisfacer sus necesidades. El análisis centrado en el usuario es una técnica muy utilizada en las metodologías orientadas a agentes, y que hace uso de la técnica de los casos de uso.

El proyecto de Cuadernos Electrónicos aspira, como se enunció en sus objetivos específicos, determinar los alcances de la metodología INGENIAS al igual que sus beneficios en el desarrollo de sistemas multiagente. Aunque la elaboración de un cuaderno electrónico para trabajar en colaboración ha sido tratado por diversas metodologías, en esta oportunidad se ha seleccionado para demostrar la efectividad de INGENIAS.

El prototipo desarrollado para un Cuaderno Electrónico comprende las áreas de Investigadores, cuadernos y proyectos de investigación. Cada investigador posee su propio cuaderno electrónico en el cual puede trabajar uno o más proyectos de investigación, permitiéndole llevar un cuaderno de anotaciones donde podrá realizar el seguimiento de sus proyectos, fechando automáticamente cada entrada realizada por el usuario.

El objetivo del cuaderno electrónico es guardar información de tipo texto, notas personales u otro tipo de datos mediante el computador, permitiendo ingresar información desde distintas ubicaciones geográficas, para hacer más eficiente y ofrecer mayor colaboración en el trabajo de equipo. Otra característica es que trabaja con XML para guardar y estructurar la información, lo que le permite agilizar el proceso de búsqueda de información.

El sistema permite la creación de un nuevo Proyecto donde el investigador ingresara el nombre de proyecto, el área a la que pertenece, que investigadores van a participar en su proyecto y tipo de financiación entre otros. También podrá seleccionar los diferentes tipos de productos que se van a realizar en su proyecto

(monografía, Artículo, tutoriales, SW) en los cuales existirá un formato estándar para su realización.

Para este proceso anteriormente descrito se encargan de llevarlo a cabo el Agente de Administración de usuarios que verifica la información de los usuarios y el Agente de Administración de Base de Datos, que realiza la creación del Proyecto. Refiriéndonos a los Productos del proyecto, de eso se encarga el Agente de Productos que es quien da la guía de cómo se deben realizar algunos de estos documentos.

Todos los investigadores en el sistema poseen un Nombre de Usuario y su respectiva contraseña. El registro lo realizara un administrador el cual será el único con permiso de realizar este proceso. El Agente Administración de BD crea el nuevo usuario y el agente de administración de usuarios colabora con dicha creación.

El agente Investigador verifica la información del usuario relacionándola y recolectándola de los Proyectos en desarrollo para ponerla a disposición de los investigadores en la entrada o pagina principal de su cuaderno y tenerla presente para cualquier interacción que el investigador desee realizar. En cuanto a la estructura del cuaderno electrónico existe un Agente Cuaderno que es el encargado de relacionar y ubicar las anotaciones al final con su respectiva fecha de entrada al sistema.

El cuaderno Electrónico usara XML en gran parte de su estructura y constitución en donde se podrá realizar identificación de la información almacenada permitiendo realizar búsquedas más eficientes. El Agente Gestión de Búsqueda es el encargado para esta labor con la colaboración del Agente Buscador.

Un caso de uso describe las posibles interacciones o usos de un usuario con el sistema. Los usuarios del sistema se denominan actores, y representan las entidades externas al sistema. El empleo de los casos de uso permite obtener los requisitos de los usuarios. A continuación se presenta el desarrollo de los casos de uso para un Cuaderno Electrónico.

Agente administrador de usuarios: Este agente se encargara de revisar los nicks de los usuarios si hay nombres repetidos y sugerir los posibles nicks validos a usar.

Agente investigador: Este agente se encarga de Recolectar toda la información relacionada con los Proyectos y mantener un fechado automático de las últimas visitas al cuaderno.

Agente Buscador: se encarga de almacenar toda información de los proyectos de forma ordenada y de realizar búsquedas organizadas.

Agente gestión de búsqueda: tiene como objetivos gestionar las búsquedas y la información a buscar.

Agente cuaderno: este agente se encarga de estructurar y administrar el cuaderno, además de fechar las visitas de los investigadores.

Agente actualización: es el encargado de hacer todas las actualizaciones periódicas al cuaderno electrónico y notificar cualquier cambio que se haya hecho.

Agente Administración Base de Datos: es el responsable de administrar la información, crear nuevos investigadores al igual q nuevos proyectos. También tiene como función consultar y actualizar la BD.

Agente Producto: tiene como función administrar los productos q ofrece el cuaderno electrónico para el trabajo en colaboración además de crear el formato del producto escogido.

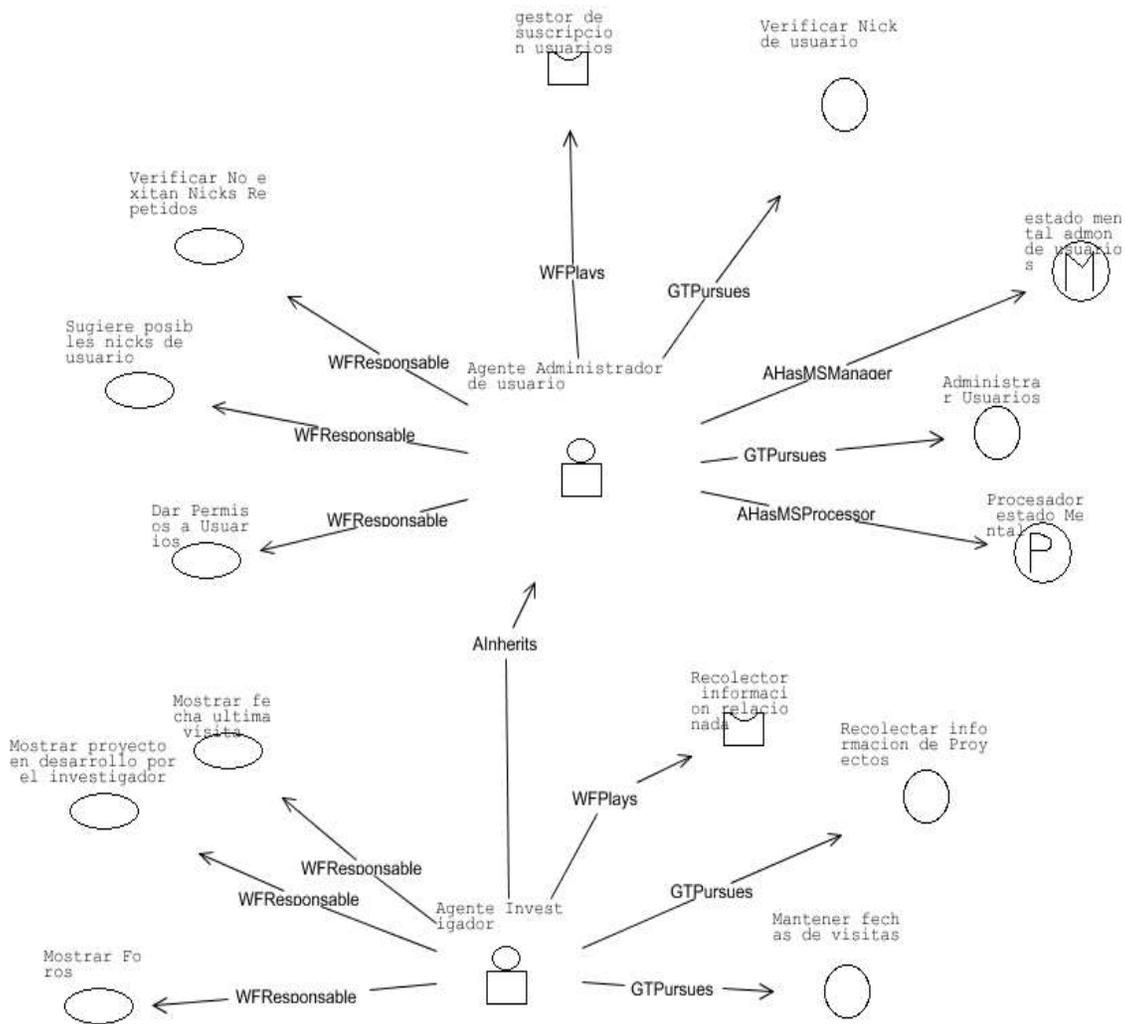
Agente asignación de permisos: Este agente gestiona, administra, asigna y verifica los permisos a investigadores para la elaboración de un proyecto específico, creando una lista de quienes pueden trabajar en colaboración con el investigador dueño de CE.

4.2 METAMODELO DE AGENTES

4.2.1 Agente Investigador y Administración de Usuarios

El Agente Administración de usuario tiene como objetivo principal, verificar que no existan nombres de usuarios repetidos además de dar permisos a otros Investigadores para trabajar en colaboración.

Figura 18. Agentes Administrador – Investigador



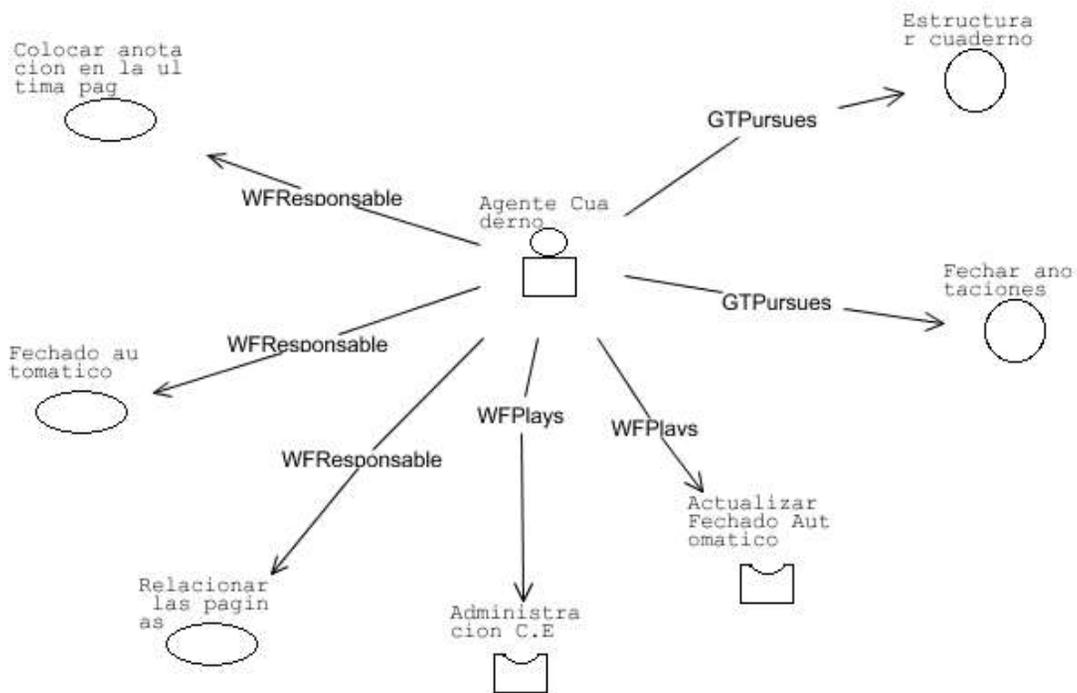
Fuente: Autores del Proyecto

El Agente Investigador es el encargado de recolectar toda la información referente al investigador dueño del cuaderno, con todos los proyectos en desarrollo, última fecha de visita y Foros existentes. Para cumplir con sus objetivos, los agentes desempeñan roles como Gestión de Suscripción Usuarios y Recolección de Información Relacionada como se muestra en la figura 18.

La figura 18 muestra que los Agentes Administrador de usuario e Investigador, tienen como objetivos principales verificar nicks, administrar los usuarios de Cuadernos Electrónicos, recolectar toda la información relacionada con los proyectos y mantener el fechado actualizado de visitas. Por medio de ellas, se busca mejorar la interacción con el usuario. En este caso el usuario al ingresar su nombre de usuario y contraseña, los agentes mostraran la fecha de la ultima visita y le concederán permiso para acceder a su cuaderno. También busca y pone a disposición los proyectos relacionados con dicho investigador ya sean propios o solo sea un participante.

4.2.2 Agente Cuaderno. El agente Cuaderno se fundamenta en Estructurar el cuaderno electrónico y fechar todas las anotaciones realizadas por el investigador. Para llevar a cabo estos objetivos el agente ejecuta los roles y una serie de tareas. Estas tareas tratan de ubicar y relacionar una anotación reciente, y ubicarla al final del documento con su respectiva fecha de entrada. Como se muestra en la figura 19, el Agente Cuaderno debe administrar toda la información relacionada con las anotaciones de los proyectos y actualizar constantemente la información para obtener un fechado real.

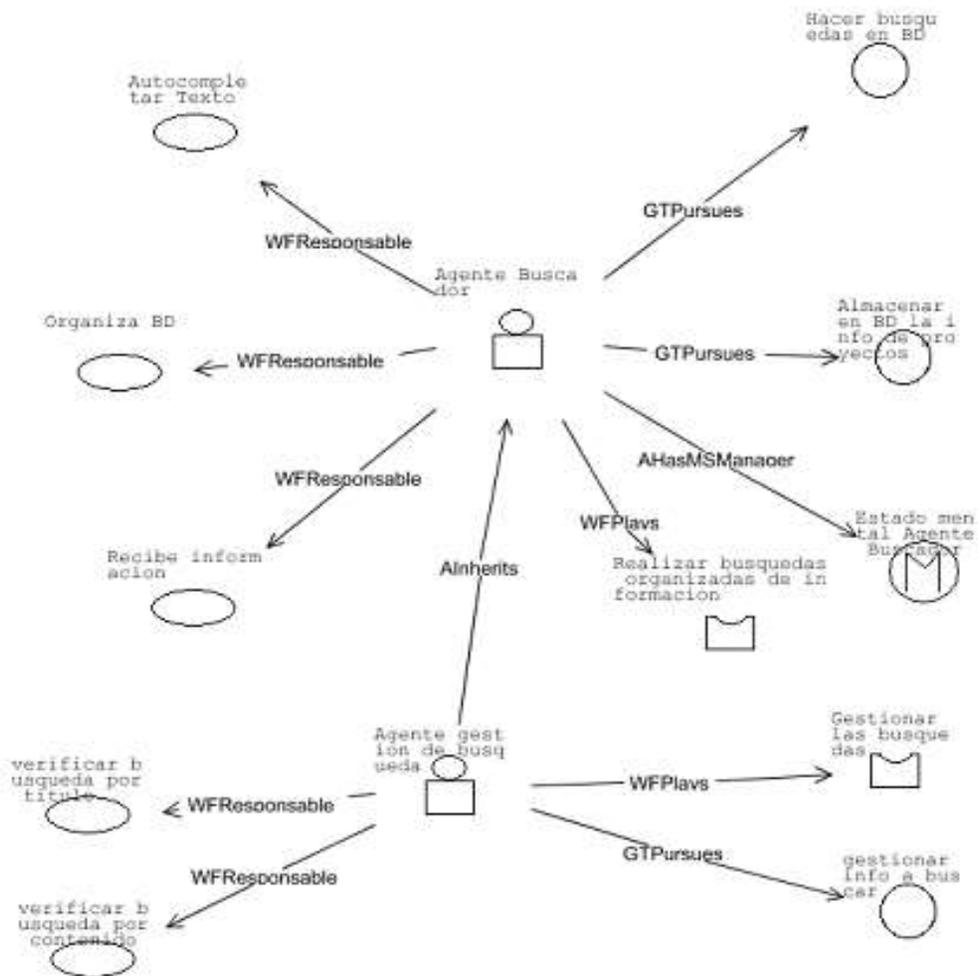
Figura 19. Agente Cuaderno Electrónico



Fuente: Autores del Proyecto

4.2.3 Agente buscador y Agente Gestión de Búsqueda. En la figura 20, el Agente Buscador tiene como responsabilidad buscar en las bases de datos la información solicitada y tener esta información organizada para que el agente de Gestión de Búsqueda pueda realizar su labor de búsquedas inteligentes y organizadas. Este Agente hace una verificación por contenido o por títulos, gestionando la búsqueda en sí.

Figura 20. Agentes Buscador y Gestión de búsqueda



Fuente: Autores del Proyecto

4.2.4 Agente Actualización. El propósito del agente Actualización (ver anexo 1) es estar pendiente de cualquier cambio que se realice en el cuaderno electrónico, cuándo el usuario ingresa y realiza una anotación. El Agente Actualización se encarga de mandar una orden de actualización, notificando que se ha realizado un cambio, para todos los participantes.

4.2.5 Agente Administración de Base de Datos. El agente administrador de BD (ver anexo 2) tiene como objetivo administrar de forma ordenada la BD del

Cuaderno Electrónico, creando un nuevo proyecto o investigador. También actualiza o consulta la Base de Datos.

Cumpliendo con sus tareas, este agente es el que se encarga de la creación de usuarios, cuando el administrador o un investigador del sistema, llena los datos además de crear, actualiza y consulta los campos necesarios de un usuario o de un proyecto

4.2.6 Agente Productos. El Agente Productos (ver anexo 3) es el responsable de administrar los productos, mostrando el formato del producto escogido y creando su respectivo documento, ya sea artículos, monografías, reporte de resultados, software.

El papel que juega este agente es en el momento de crear un proyecto, el recibe los datos de que productos se van a realizar y al poseer esta información con la ayuda de XML, el agente puede establecer como debe ir constituido dicho documento, con el fin de ayudar al usuario en el desarrollo de este.

4.2.7 Agente recordatorio. El agente Recordatorio (ver anexo 4) es el encargado de mostrar recordatorios de próximas fechas de entrega de proyectos.

4.2.8 Agente asignación de permisos. El Agente Asignación de Permisos (ver anexo 5) se encarga de crear, asignar y verificar una lista de permisos de investigadores participantes, en cada uno de los proyectos, permitiéndoles trabajar en colaboración con el investigador dueño del proyecto.

Tabla 1. Instancias de entidades de meta-modelo de agentes

ENTIDAD	ENTIDAD DEL META-MODELO	DOCUMENTACIÓN
Verificar nick de usuario	Objetivo	Este objetivo se alcanza cuando se satisface la tarea de verificación de usuarios al registrarse.
Administrar usuarios	Objetivo	Este objetivo administra a todos los usuarios q trabajan en colaboración en el cuaderno electrónico.
Recolectar Información de usuarios	Objetivo	Este objetivo guarda toda la información relacionada con los usuarios
Mantener fechas de visitas	Objetivo	Este objetivo es alcanzado de forma automática cuando el investigador hace sus anotaciones.
Agente administrador de usuario	Agente	Este agentes el encargado de hacer la verificación de nicks y administración de usuarios
Agente investigador	Agente	Se encarga de mantener el fechado automático de la ultima entrada al cuaderno además de guardar toda la información
Agente buscador	Agente	Este agente se encarga de hacer las búsquedas de proyectos y la información y almacenarlos.
Agente gestión de búsqueda	Agente	Se encarga de gestionar las búsquedas de forma organizada

Verificar no existan nicks repetidos	Tarea	Cada vez que se necesite materia prima el agente debe utilizar las formulas para el calculo optimo de la mercancía.
Dar permisos de usuarios	Tarea	Verifica la orden recibida con su stock para enviarla al depto de producción o ventas.
Mostrar fecha última visita	Tarea	Se encarga de informar al usuario final cual es el mejor proveedor de acuerdo a la distancia respecto a la empresa y los precios mas apropiados para el pedido solicitado.
Mostrar proyectos en desarrollo	Tarea	Se confirma si el proveedor envió el pedido solicitado.
Mostrar foros	Tarea	El agente Recursos recibe y verifica que cada articulo enviado por el proveedor este en buen estado.
Recolectar información relacionada	Rol	Los artículos en buen estado serán registrados en la base de datos.
Gestor de suscripción usuarios	Rol	La materia prima solicitada por el departamento de Producción para hacer un artículo con esta.
Administración de usuarios	Gestor Estado mental	Una vez creado los usuarios, este estado mental administra los usuarios
Procesador estado	Procesador estado	Se trata de procesar la

mental	mental	administración de usuarios
Hacer búsquedas	Objetivo	Este objetivo se cumple cuando se hace la gestión de búsqueda de la información
Almacenar la información de proyectos	Objetivo	Este objetivo es alcanzado cuando la información es guardada de forma organizada
Gestionar la información a buscar	Objetivo	Este objetivo se cumple cuando se hace una verificación de la información ya sea por contenido o por contenido
Estructurar cuaderno	Objetivo	Se cumple cuando el agente cuaderno guarda la información y estructura el cuaderno
Fechar anotaciones	Objetivo	Se cumple cuando se hacen las anotaciones en el cuaderno
Realizar búsquedas organizadas de información	Rol	Es el encargado de gestionar las búsquedas de forma organizada
Gestionar información a buscar	Rol	Es el encargado de gestionar las búsquedas de información
Estado mental agente buscador	Estado mental	Una vez la información es guardada, ésta es buscada y administrada
Organiza la Base de Datos	Tarea	Se encarga de la organización de la información

Recibe la información	Tarea	Se encarga de recibir toda la información y guardarla
Verificar búsquedas	Tarea	Se encarga de hacer la verificación de búsquedas de la información en los proyectos.
Poner anotación en la última página	Tarea	Se encarga de que la ultima anotación hecha sea de forma ordenada
Relacionar páginas	Tarea	Se encarga de relacionar las páginas del cuaderno.

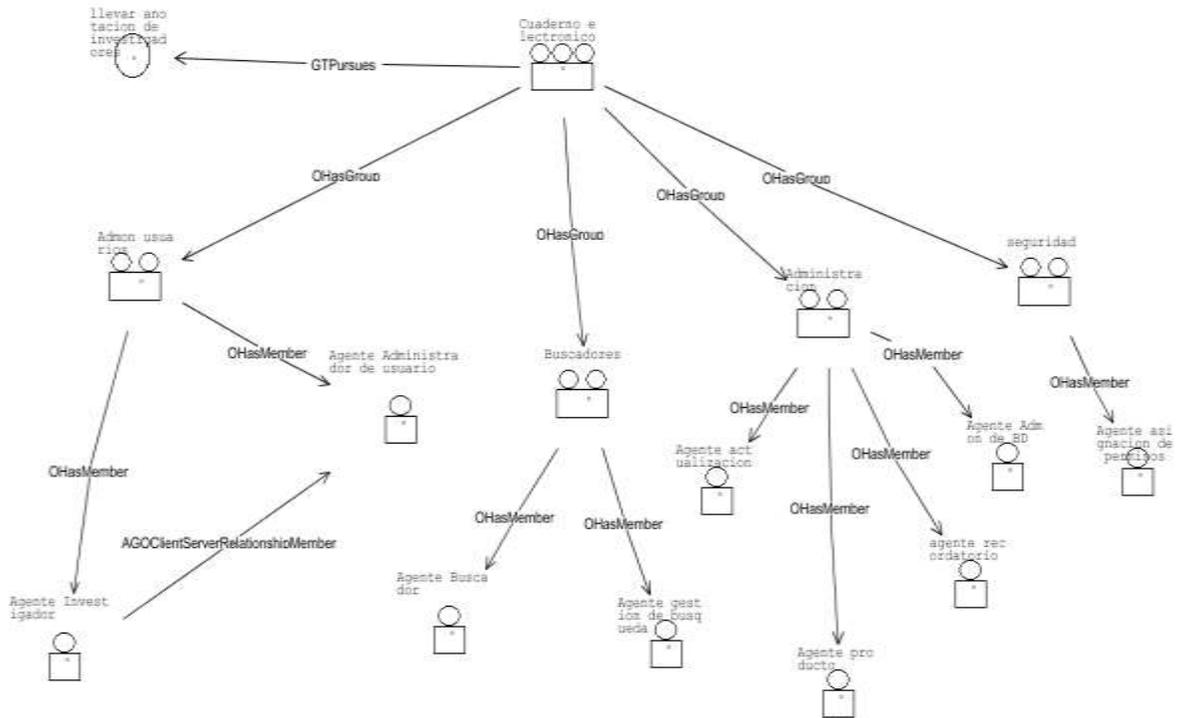
Fuente: Autores del Proyecto

4.3 META-MODELO DE ORGANIZACIÓN

En la figura 26, se puede observar que la organización del cuaderno electrónico tiene como objetivo principal *llevar las anotaciones del investigador* y esta conformada por entidades de agentes encargados de la administración de Usuarios y el manejo de la información. Dichos agentes se encuentran estructurados dentro de grupos, como se puede apreciar en la figura.

El primer grupo llamado *Admón. De usuarios* lo conforman los agentes de investigador y de admón. de usuarios, que se relacionan entre si para dar cumplimiento a sus tareas. Otro grupo *buscadores* esta conformado por los agentes búsqueda y gestión de búsqueda que son los responsables de organizar la información y de las búsquedas inteligentes de información ya sea por titulo o por contenido.

Figura 26. Meta-modelo de Organización



Fuente: Autores del Proyecto

La administración es responsable el agente *admón. BD* que se encarga de la creación de proyectos como de usuarios. Pero en este proceso también intervienen otros agentes que complementan este proceso que son los agentes de actualización, de productos y el de recordatorio que controlan los cambios realizados en el cuaderno y establecen formatos para los productos.

Por ultimo se encuentra en el *grupo de seguridad* el agente asignación de permisos que realiza la labor de dar acceso a los investigadores a determinados

proyectos de acuerdo al permiso concebido por el Investigador dueño del proyecto.

Tabla 2. Instancias de entidades de meta-modelo de organización

ENTIDAD	ENTIDAD DEL META-MODELO	DOCUMENTACIÓN
Llevar anotación de investigadores	Objetivo	Todas las anotaciones de los investigadores se deben llevar de forma ordenada
Administración de usuarios	Grupo	Este grupo es el encargado de la administración de los usuarios registrados en el sistema.
Buscadores	Grupo	Este grupo es el encargado de gestionar las búsquedas.
Administración	Grupo	Este grupo es el encargado de la administración de los cuadernos.
Agente administrador de usuario	Agente	Este agente es el encargado de la administración de los usuarios.
Agente investigador	Agente	Este agente se encarga de guardar y manejar toda la información del investigador
Agente buscador	Agente	Este agente se encarga de hacer las búsquedas de proyectos y la información.
Agente gestión de	Agente	Este agente se encarga de gestionar las

búsqueda		búsquedas de información de forma organizada
Seguridad	Grupo	Ese grupo es el encargado de la seguridad del cuaderno
Agente actualización	Agente	Este agente es el encargado de hacer las actualizaciones del cuaderno electrónico
Agente producto	Agente	Este agente se encarga de de administrar todos los productos y servicios que ofrece el cuaderno electrónico.
Agente administración Base de Datos	Agente	Este agente se encarga de administrar toda la información que maneja el cuaderno.
Agente asignación de permisos	Agente	Este agente se encarga de la asignación de permisos para el trabajo en colaboración

Fuente: Autores del Proyecto

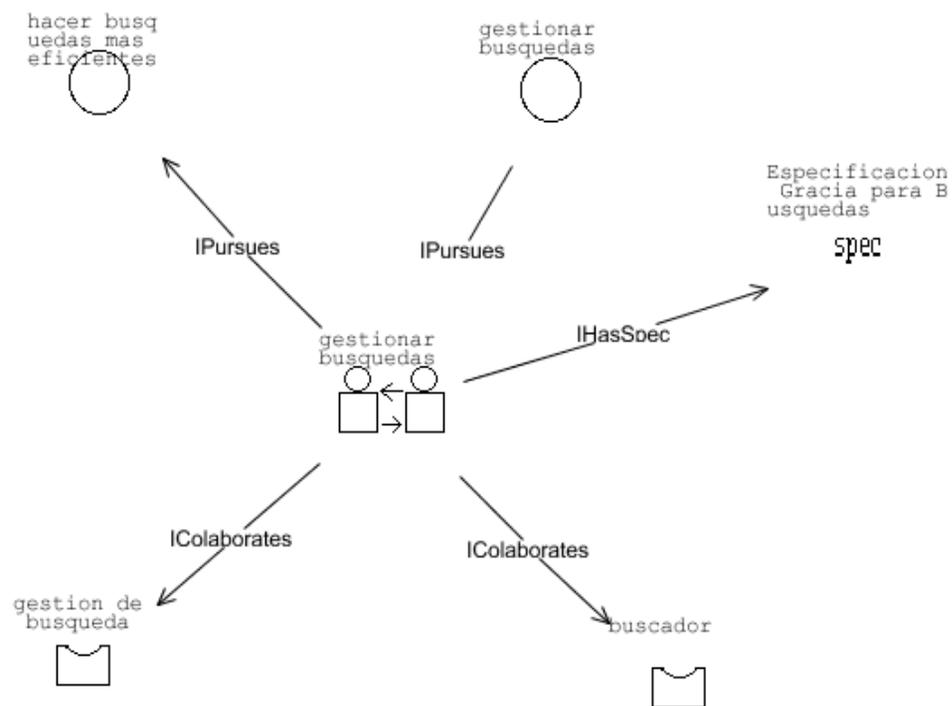
4.4 META-MODELO DE INTERACCIÓN

El meta-modelo de interacción busca especificar quien ejecuta las interacciones y como estas afectan al sistema, que tareas se ejecutan en la interacción y que tareas producen dicha interacción.

4.4.1 Interacción Gestionar búsquedas. En la figura 27, se tiene la interacción *Gestionar Búsquedas*, que esta encargada de la consulta de cierta información.

Esta situación se da cuando el usuario desea buscar cierta información ya sea por título o por contenido, los agentes que colaboran con el proceso se encargan de suministrar dicha información que debe estar debidamente estructurada y organizada para que la interacción se lleve a cabo.

Figura 27. Gestión de Búsquedas

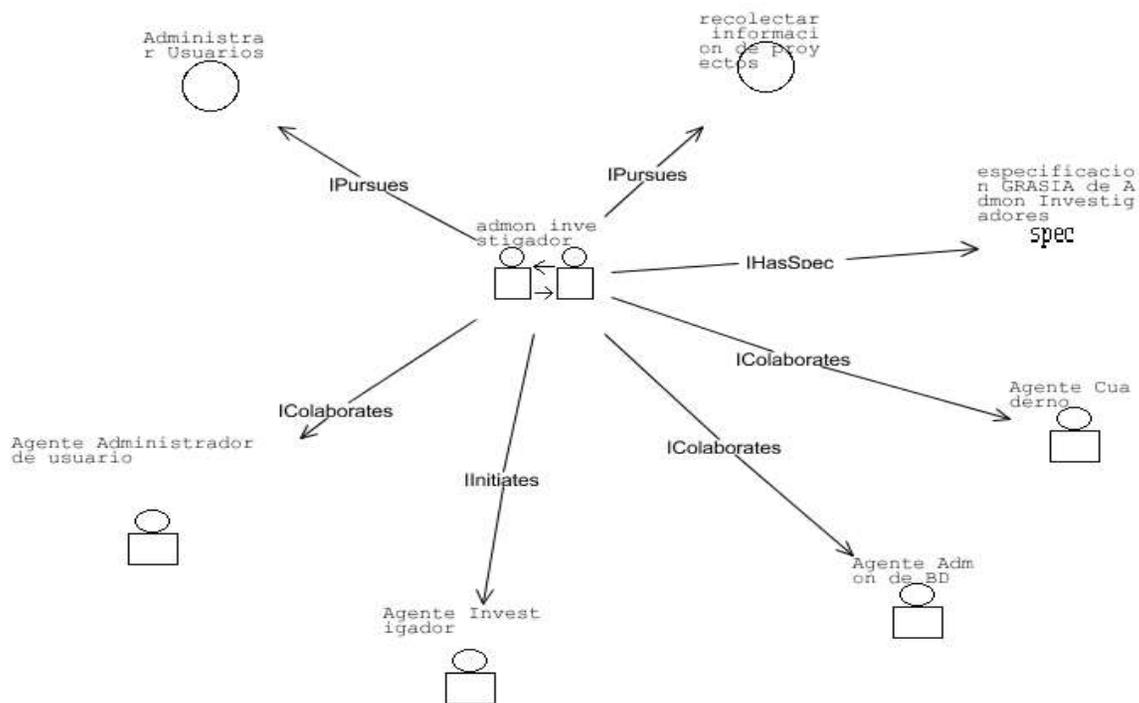


Fuente: Autores del Proyecto

4.4.2 Interacción Administración e Investigador. En la figura 28, la administración de investigador se busca la recolección de la información y la administración de usuarios, esta interacción la inicializa el agente investigador de usuario pues este es el encargado de recolectar información de los proyectos de los investigadores y le colabora el agente de admón. De usuarios pero en el proceso de la

administración no solo intervienen estos agentes también los agentes admón. de base de datos y el agente cuaderno que complementan este proceso creando nuevos usuarios y relacionando la información con los investigadores.

Figura 28. Interacción administración con Investigador



Fuente: Autores del Proyecto

4.4.3 Interacción gestión de permisos. En esta interacción se maneja la asignación de los permisos, la cual es inicializada por el Agente Asignación de permisos (ver anexo 6) cuando un usuario ingresa a su cuaderno o a determinado proyecto, comprobando que tipo de permiso posee. Los agentes de Administración de Base de Datos y el Administración de usuario colaboran para

realizar esta comprobación para dejar que los investigadores interactúen y accedan al cuaderno de acuerdo con su tipo de permiso.

4.4.4 Interacción actualización de C. E. En esta interacción de actualización del Cuaderno Electrónico (ver anexo 7), el agente actualización es el encargado de inicializar el proceso notificando cualquier tipo de cambio en el cuaderno electrónico realizado por el investigador colaborándole están los agentes de cuaderno y el de admón. de BD. Estructurando la información que se va a anexando al cuaderno.

Se muestra la interacción para llevar acabo la actualización continua del Cuaderno Electrónico.

4.4.5 Interacción de Gestión de recordatorios. En la realización de los recordatorios (ver anexo 8) se crea una interacción donde es inicializada por el agente recordatorio donde van a estar explicitas las fechas de posible entrega de los proyectos dándole aviso al investigador que la fecha se aproxima y para esta labor también le colaborara el agente de BD puesto que posee la información almacenada de y relacionada de los proyectos con sus respectivas fechas.

Tabla 3. Instancias de entidades de meta-modelo de interacción

ENTIDAD	ENTIDAD DEL META-MODELO	DOCUMENTACIÓN
Especificación GRACIA Investigadores	Especificación GRASIA	
Especificación	Especificación	

GRACIA Búsquedas	GRASIA	
Especificación GRACIA Actualización	Especificación GRASIA	
Gestionar búsquedas	Interacción	Gestionar búsquedas se produce cuando hay una solicitud de búsqueda de información.
Gestionar Admón. investigador	Interacción	Gestionar Admón. investigador se produce cuando se hace cuando se administra investigadores
Gestionar permisos	Interacción	Gestionar permisos se produce cuando se establecen los permisos a los investigadores para el trabajo en colaboración
Agente administración Base de Datos	Agente	Este agente se encarga de administrar toda la información que maneja el cuaderno.
Agente asignación de permisos	Agente	Este agente se encarga de la asignación de permisos para el trabajo en colaboración

Fuente: Autores del Proyecto

4.5 META-MODELO DE OBJETIVOS Y TAREAS

En este meta-modelo se plantean todos los objetivos y tareas requeridas para cumplir con los objetivos propuestos y para el funcionamiento del sistema.

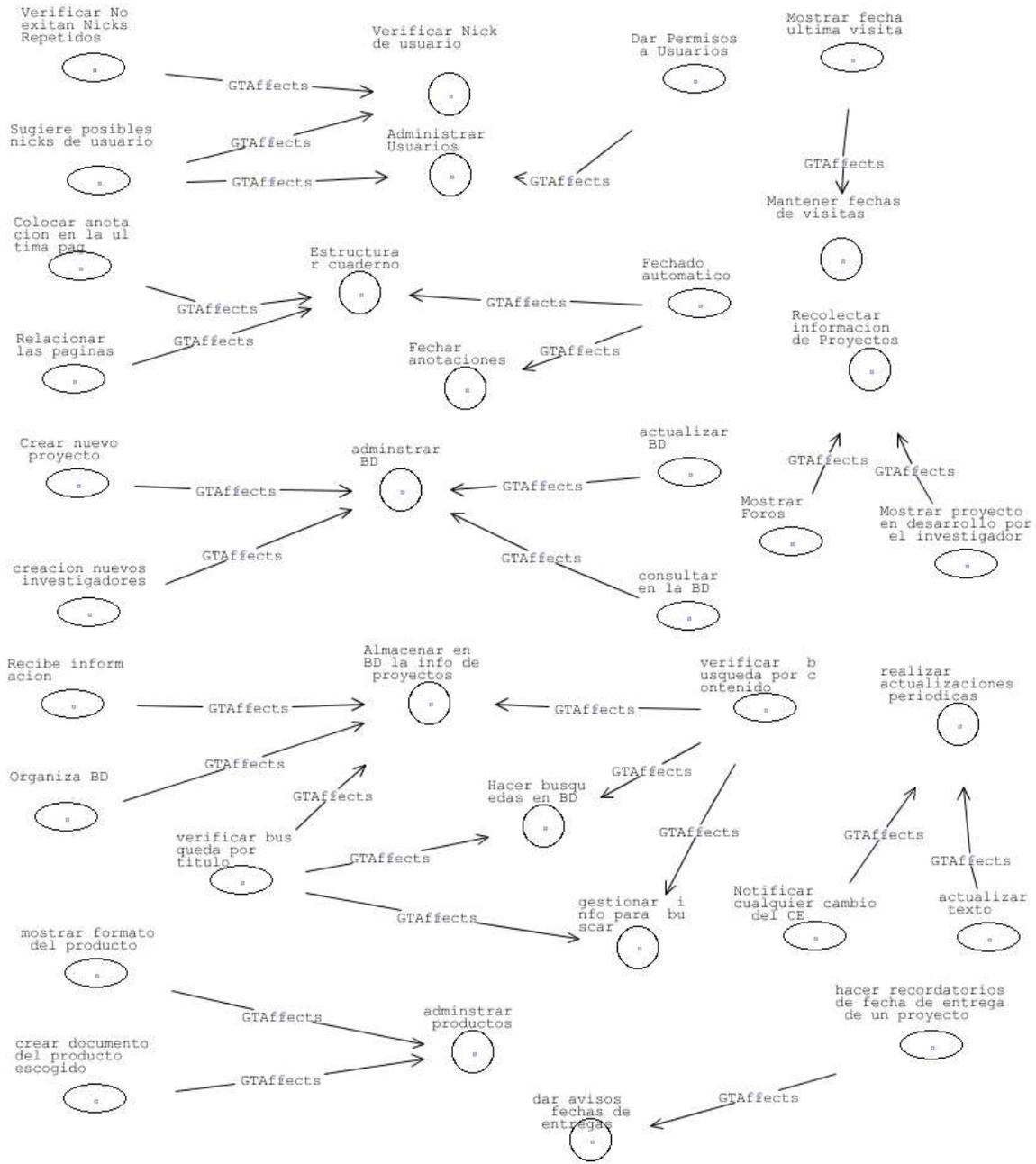
En la figura 32, se puede observar las relaciones existentes entre las tareas y determinados objetivos para llevarlos a un estado satisfactorio o llevarlos a que fallen, estas relaciones se encuentran separadas de acuerdo al proceso que se realice. Si es la creación de un nuevo usuario, muestra qué tareas intervienen en dicha creación y que objetivos son alcanzados y a cuales otros afecta.

Hablando del cumplimiento o el fracaso de dichos objetivos se realizó un diagrama mas detallados de cuales tareas son esenciales que pueden llevar a fallar un determinado objetivo si estas no cumplen su labor (ver anexo 9).

Es importante saber los flujos de trabajo de las tareas y que aplicaciones van relacionadas con ellas. Para esto se realizó un diagrama de Objetivos y tareas mas detallado (ver anexo 10), en donde cada tarea se muestra como interactúa y que elementos necesita, que produce dicho flujo.

Mas adelante se podrá observar los flujos de las diferentes tareas (ver anexo 11), un claro ejemplo es la tarea *Mostrar proyecto en desarrollo por el investigador*, consume información de investigadores, como de proyectos la cual como producto de la ejecución de esta tarea se vuelve una lista de proyectos en los cuales va a participar cada investigador.

Figura 32. Satisfacción de Objetivos



Fuente: Autores del Proyecto

Tabla 4. Instancias de entidades de meta-modelo de objetivos y tareas

ENTIDAD	ENTIDAD DEL META-MODELO	DOCUMENTACIÓN
Organiza la Base de Datos	Tarea	Se encarga de la organización de la información
Recibe la información	Tarea	Se encarga de recibir toda la información y guardarla
Verificar búsquedas	Tarea	Se encarga de hacer la verificación de búsquedas de la información en los proyectos.
Poner anotación en la última página	Tarea	Se encarga de que la ultima anotación hecha sea de forma ordenada
Relacionar páginas	Tarea	Se encarga de relacionar las páginas del cuaderno.
Dar permisos a usuarios	Tarea	Se encarga de conceder permisos a un investigador para trabajar en colaboración.
Crear nuevo usuario	Tarea	Tiene como función la creación de nuevos usuarios
Mostrar foros	Tarea	Se encarga de la creación de todos los foros hechos por los investigadores
Mostrar proyectos	Tarea	Se encarga de mostrar los proyectos en desarrollo por el investigador

Almacenar la información de proyectos	Objetivo	Este objetivo es alcanzado cuando la información es guardada de forma organizada
Gestionar la información buscar	Objetivo	Este objetivo se cumple cuando se hace una verificación de la información ya sea por contenido o por contenido
Estructurar cuaderno	Objetivo	Se cumple cuando el agente cuaderno guarda la información y estructura el cuaderno
Fechar anotaciones	Objetivo	Se cumple cuando se hacen las anotaciones en el cuaderno
Recolectar información de proyectos	Objetivo	Se cumple cuando se guarda toda la información relacionada con cada uno de los proyectos realizados por los investigadores
Hacer búsquedas	Objetivo	Este objetivo se cumple cuando se hace la gestión de búsqueda de la información de los proyectos y foros hechos por los investigadores

Fuente: Autores del Proyecto

4.6 METAMODELO DE ENTORNO

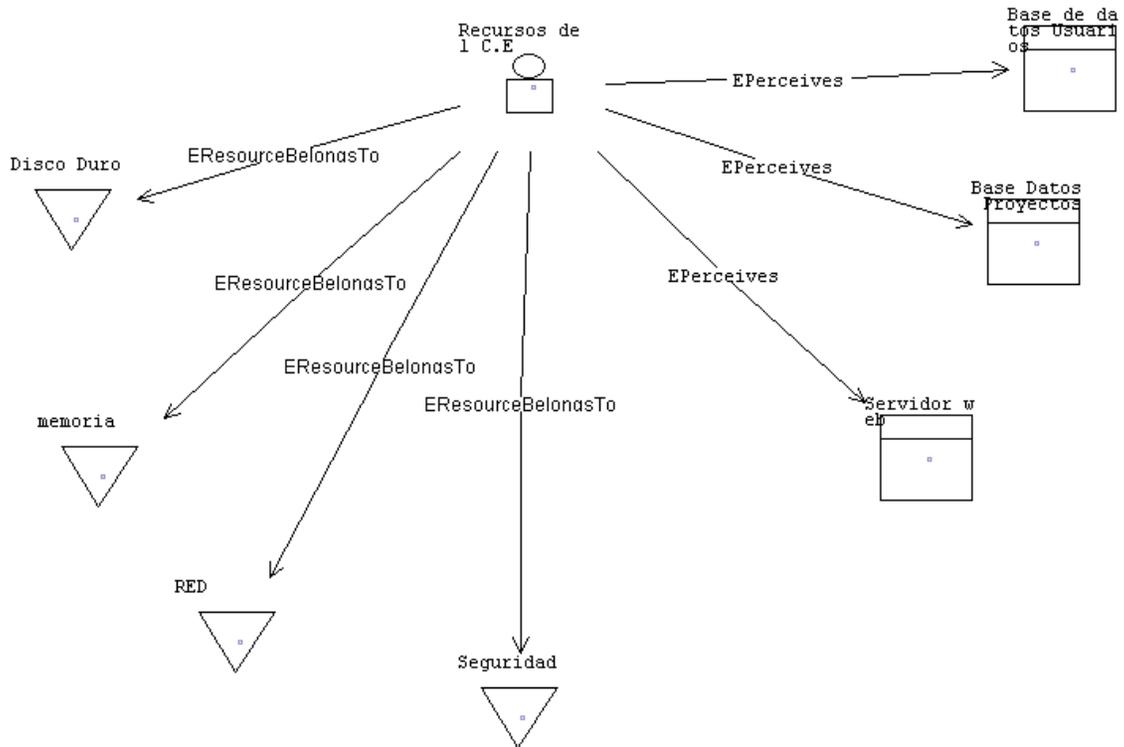
En la figura 36, el Meta-modelo de Entorno empieza restringiendo el tipo de elementos que van a aparecer en el sistema, se distinguen tres posibles tipos de elementos: agentes, recursos y aplicaciones.

- Recursos: objetos del entorno que no proporcionan una funcionalidad concreta, pero son indispensables para la ejecución de las tareas, cuyo uso es restringido a consumir o restituir.
- Aplicación: cuando un recurso sea más complejo, como una base de datos.
- Agente: el encargado de usar los recursos y aplicaciones.

A lo largo del desarrollo se identificaron elementos necesarios para que las tareas se puedan llevar a cabo. Estos elementos son computacionales con características cercanas a los objetos. Estos elementos han sido reunidos aquí, distinguiendo entre aplicaciones que se crean junto con el sistema, que se suponen ya desarrolladas (admón. de usuarios y Proyectos), y del entorno percibe base de datos de usuarios, base de datos de proyectos y servidor Web. También se han añadido recursos más convencionales, en concreto, memoria, disco duro y conexión a la red.

En la figura 36, se muestra los recursos necesarios del cuaderno electrónico que se van a crear archivos de cada investigador como del cuaderno en general, Los recursos como memoria, disco duro, conexión a la red y seguridad se tienen en cuenta ya que son requeridos para el funcionamiento del sistema.

Figura 36. Recursos del Cuaderno Electrónico



Fuente: Autores del Proyecto

Tabla 5. Instancias de entidades de meta-modelo de entorno

ENTIDAD	ENTIDAD DEL META-MODELO	DOCUMENTACIÓN
Disco Duro	Recurso	Describe los requerimientos en cuanto a espacio de almacenamiento, necesario para que funcione el sistema
Red	Recurso	Describe cual es la conexión

		mínima de la red para la comunicación de agentes entre las maquinas.
Memoria	Recurso	Describe la memoria requerida para que la aplicación funcione de manera correcta
Seguridad	Recurso	Sirve para que no todos los investigadores tengan acceso a proyectos personales y catalogar diferentes perfiles de usuario
Recursos del C.E	Agente	Se encarga de relacionar las páginas del cuaderno.
Archivos usuarios	Aplicación	Realiza labores de relacionadas con los usuarios, manejo de los productos terminados y todo lo relacionado con las mejores practicas de ventas
Archivos proyectos	Aplicación	Archivo utilizado en el sistema para la gestión de datos. Se reaprovechara para almacenar el estado del sistema

Fuente: Autores del proyecto

5. FASE DE IMPLEMENTACION

5.1 HERRAMIENTAS NECESARIAS PARA LA IMPLEMENTACION

En primera instancia es necesario tener instaladas las siguientes herramientas para la codificación, compilación, depuración de los agentes e interfaces. También se necesita una base de datos, un servidor Web. Las herramientas utilizadas son:

- JAVA jsdk 1.4._06.
- Ingenias IDE 2.0, se utilizara como herramienta de soporte para generar los diagramas de casos de uso, y representar las actividades y entregas de las fases de análisis y diseño cada uno de los meta-modelos.
- JDom, JCreator LE
- Después de haber instalado el software anteriormente mencionado es importante descargar un servidor Web y un motor de bases de datos. En nuestro proyecto utilizaremos un servidor Apache http server 2.0 y Jakara Apache Tomcat.
- Para la creación del cuaderno se piensa usar JSP y JAVA para la creación de los agentes.
- Para la creación de los archivos de usuario y del cuaderno se manejaran con XML programados en JAVA y JSP.

5.2 DESARROLLO CON XML

Para la creación de estos documentos fue necesaria la creación de definición de tipo de documentos (DTD), los cuales establecen cual va a ser la estructura del archivo y los elementos mínimos que debe contener cada archivo de xml.

5.2.1 DTD Usuarios. A los usuarios se les pide una serie de datos mínimos por llenar como el ID que es con el que se va a identificar el usuario, nombres y apellidos del investigador, que nivel de estudio tiene (universitario, postgrado, doctorado) a que área de conocimiento pertenece.

<!ELEMENT USUARIO

(IdUsuario+,Nombre+,Apellido+,NivelEstudio+,AreasConocimiento+,Email+,NombreUsuario+,Contraseña+)>

<!ELEMENT IdUsuario (#PCDATA)>

<!ELEMENT Nombre (#PCDATA)>

<!ELEMENT Apellido (#PCDATA)>

<!ELEMENT NivelEstudio (#PCDATA)>

<!ELEMENT AreasConocimiento (#PCDATA)>

<!ELEMENT Email (#PCDATA)>

<!ELEMENT NombreUsuario (#PCDATA)>

<!ELEMENT Contraseña (#PCDATA)>

5.2.2 DTD Cuaderno. Los cuadernos de cada uno de los investigadores estarán relacionados con la información mostrada en el código, con esto será posible identificar a que proyecto pertenece a cada investigador mostrando el nombre del usuario que maneja el cuaderno y su identificador al igual que el identificador del cuaderno.

<!ELEMENT CUADERNO (NombreUsuario+,IdUsuario+,IdCuaderno+)>

<!ELEMENT NombreUsuario (#PCDATA)>

<!ELEMENT IdUsuario (#PCDATA)>

<!ELEMENT IdCuaderno (#PCDATA)>

5.2.3 DTD Anotación. Las anotaciones van a llevar la información que el usuario dueño del cuaderno haga durante el desarrollo de un proyecto determinado en el que esté trabajando. El DTD de anotaciones lleva una serie de información vital para su buena función como es el id del proyecto al cual va a ser realizada dicha anotación, el id del usuario que es quien hace la anotación con su respectivo nombre, el título y contenido de la anotación.

```
<!ELEMENT ANOTACION (IdUsuario+,Titulo+,Contenido+,NombreUsuario+,
IdProyecto+)>
  <!ELEMENT IdUsuario (#PCDATA)>
  <!ELEMENT Nombre (#PCDATA)>
  <!ELEMENT Titulo (#PCDATA)>
  <!ELEMENT Contenido (#PCDATA)>
  <!ELEMENT NombreUsuario (#PCDATA)>
  <!ELEMENT IdProyecto (#PCDATA)>
```

5.2.4 DTD Proyectos. A la hora de crear un proyecto estos son los campos mínimos que el investigador debe llenar para su efectiva creación. El nombre del nuevo proyecto que se está creando, el grupo de investigación al cual pertenece, que entidades van a financiar el proyecto, a que convocatoria pertenece, que tipos de productos va a utilizar, relaciona el identificador del investigador creador del proyecto con su respectivo idProyecto, además de una breve descripción de que va a tratar el proyecto.

```
<!ELEMENT
PROYECTOS(NombreProyecto+,Investigadores+,GrupoInvestigacion+,Financiacion+,Convocatoria+,Productos+,IdInvestigador+,IdProyecto+,Descripcion+)>
  <!ELEMENT NombreProyecto (#PCDATA)>
```

```
<!ELEMENT Investigadores (#PCDATA)>
<!ELEMENT GrupoInvestigacion (#PCDATA)>
<!ELEMENT Financiacion (#PCDATA)>
<!ELEMENT Convocatoria (#PCDATA)>
<!ELEMENT Productos (#PCDATA)>
<!ELEMENT IdInvestigador (#PCDATA)>
<!ELEMENT IdProyecto (#PCDATA)>
<!ELEMENT Descripcion (#PCDATA)>
```

5.2.5 DTD Permisos de Usuarios. Para el manejo de los permisos se va a trabajar una tabla de permisos donde existirán los identificadores de los investigadores con sus respectivos permisos para cada proyecto. El DTD permisos de usuarios consta del id del proyecto en el que se esta trabajando y el id del usuario que esta trabajando en colaboración con el investigador del proyecto, que en este caso es quien permite la entrada a la información pero no permite modificar el proyecto en sí.

```
<!ELEMENT PERMISOSUSUARIOS
(IdProyecto+,IdInvestigador+,TipoPermiso+)>
  <!ELEMENT IdInvestigador (#PCDATA)>
  <!ELEMENT IdProyecto (#PCDATA)>
  <!ELEMENT TipoPermiso (#PCDATA)>
```

5.2.6 DTD Artículo. Los Artículos que se piensen realizar en el cuaderno o en determinado proyecto contienen elementos tales como el nombre del artículo con su respectivo identificador, contenido y resumen el proyecto, palabras claves con las que se identifica el artículo, introducción, conclusiones y referencias bibliográficas, además del área de conocimiento a la cual pertenece.

```
<!ELEMENT ARTICULO (NombreArticulo+,IdArticulo+,Contenido+,Resumen+,
PalabrasClave+, Introduccion+ Conclusiones+, Referencias+,
AreasConocimiento+, Titulo+)>
```

```
  <!ELEMENT NombreArticulo (#PCDATA)>
  <!ELEMENT IdArticulo (#PCDATA)>
  <!ELEMENT Contenido (#PCDATA)>
  <!ELEMENT Resumen (#PCDATA)>
  <!ELEMENT PalabrasClave (#PCDATA)>
  <!ELEMENT Introduccion (#PCDATA)>
  <!ELEMENT Conclusiones (#PCDATA)>
  <!ELEMENT Referencias (#PCDATA)>
  <!ELEMENT AreasConocimiento (#PCDATA)>
  <!ELEMENT Titulo (#PCDATA)>
```

5.2.7 DTD Monografía. Las monografías como mínimo deberán poseer información como el nombre y el identificador de la monografía, título, contenido y el identificador del investigador.

```
<!ELEMENT MONOGRAFIA (NombreMonografia+,IdMonografia+,Contenido+,
Titulo+, IdInvestigador+, Capitulo+)>
```

```
  <!ELEMENT NombreMonografia (#PCDATA)>
  <!ELEMENT IdMonografia (#PCDATA)>
  <!ELEMENT Contenido (#PCDATA)>
  <!ELEMENT Titulo (#PCDATA)>
  <!ELEMENT IdInvestigador (#PCDATA)>
  <!ELEMENT Capitulo (#PCDATA)>
```

5.3 INTERFAZ GRAFICA

A continuación se explicara parte de la interfaz grafica del cuaderno electrónico.

Figura 37. Pagina Principal del C.E

Breve descripción de cuadernos electrónicos

diversos países. A partir de este prototipo, ha surgido gran cantidad de proyectos de Cuadernos Electrónicos en distintas organizaciones, Instituciones educativas, en el campo de la medicina, grupos de investigación y para uso personal. Basados en la arquitectura del prototipo ORNL se desarrollará el Cuaderno Electrónico teniendo en cuenta algunos aspectos importantes como la toma de anotaciones, seguridad, administración de sesión e interfaces.

El investigador debe digitar su nombre

BIENVENIDOS AL CUADERNO ELECTRONICO FAVOR LOGEARSE

Nombreusuario:

Contraseña:

Entrar **Dirige al cuaderno del investigador**

ENGINE

- CE Engine N
- CE Engine 2
- CE Engine 1
- Admón. Sesión
- Seguridad
- Admón. Base datos

Software bus

INTERFACES

- Navegador Web
- Área de investigación X1 del cuaderno

Fuente: Autores del Proyecto

En la figura 37, nos encontramos en la página principal del cuaderno donde el investigador podrá entrar con su propio nombre de usuario y contraseña. En esta página, el investigador encontrará una definición muy completa acerca de

cuadernos electrónicos. Al entrar al cuaderno los agentes de admón. de BD y investigadores intervienen en este proceso para buscar y poner a disposición la información relacionada con el investigador.

Figura 38. Nuevo proyecto

UNAB
BUCARAMANGA

NUEVO PROYECTO

INGRESE LOS DATOS DEL PROYECTO

Nombre del Proyecto: REDES INALÁMBRICA

Id Proyecto: 11 — Generado automático

Id Usuario: 13744340 — Lo captura del inicio de sesión

Grupo de Investigación: Ciencias Aplicadas

Financiación: ICETEX-COLCIENCIAS — Lista para escoger la opción

Convocatoria: Plan Bienal — Lista para escoger la opción

Productos:

Artículo Monografía Software

Descripción: Descripción del proyecto

Las redes inalámbricas pueden tener un gran auge debido a la necesidad de movimiento que se requiere debido a los cambios de ritmo de vida que se sufre a través del tiempo, esta tecnología puede ser utilizada

Crear Proyecto — Creación del nuevo proyecto

Fuente: Autores del Proyecto

En la figura 39, al realizar una anotación en el cuaderno, se encontrara un cuadro de texto para guardar las anotaciones que el investigador desee para cada uno de sus proyectos. Este cuadro de texto Tendrá la opción de seleccionar el tipo de letra, color, alineación, tamaño entre otras. Al ingresar esta información al sistema, automáticamente se le anexara una fecha a la anotación y se pondrá al final del cuaderno. El investigador que esté creando la anotación tendrá q digitar el nombre de la anotación. Por otra parte, el identificador del usuario y su nombre aparecerá automáticamente desde el inicio de sesión y el identificador del proyecto será generado automáticamente por el sistema.

Figura 40. Nuevo Investigador

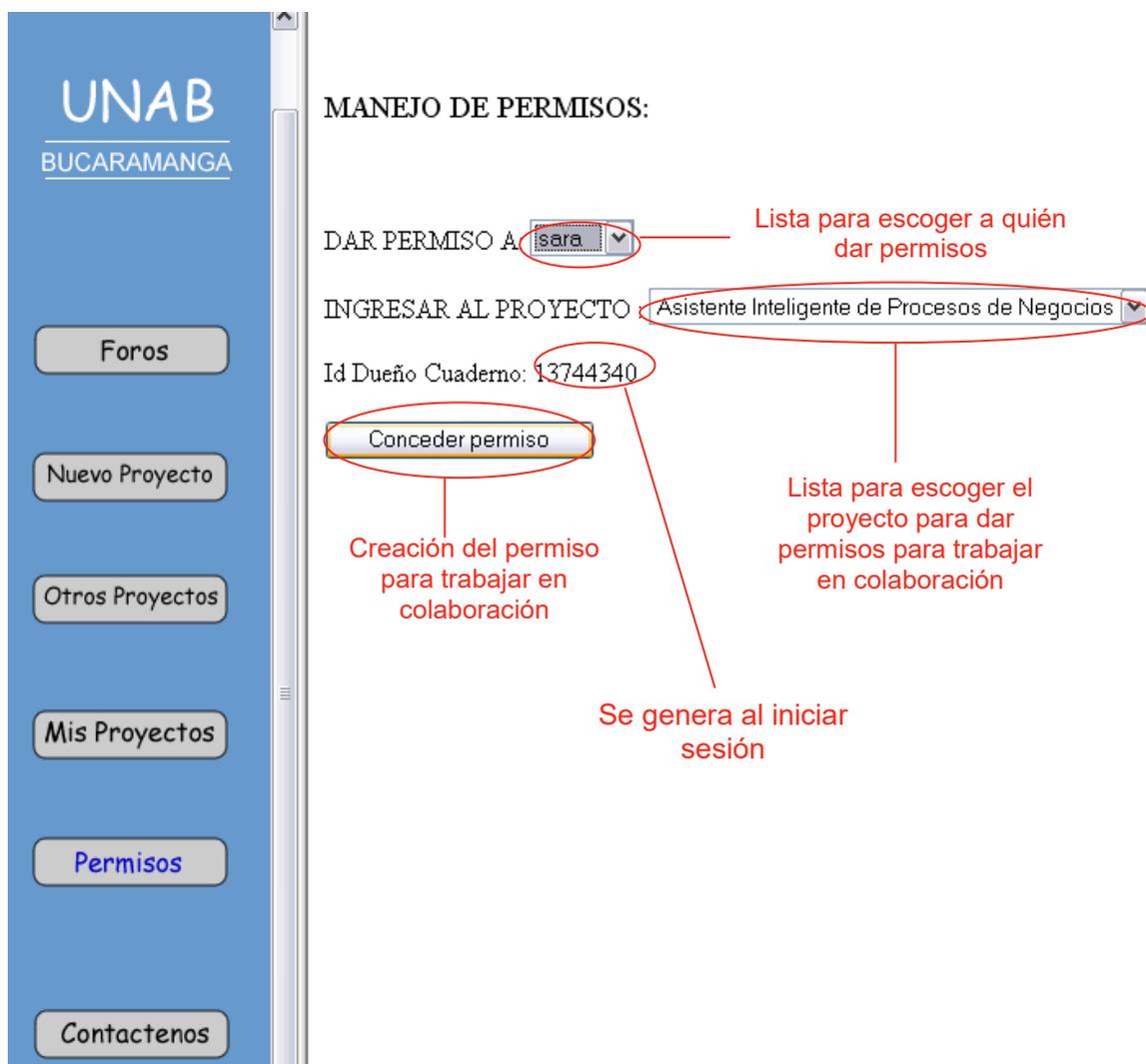
The image shows a web interface for creating a new investigator. On the left is a blue sidebar with three buttons: 'Proyectos en desarrollo', 'Nuevo Investigador', and 'Contadores'. The main area is titled 'ADMINISTRADOR' and contains a registration form with the following fields: 'Nombres de Investigador', 'Apellidos de Investigador', 'Nivel de estudio', 'Áreas de conocimiento', 'EMail', 'Identificación', 'Nombre de Usuario', and 'Contraseña'. A 'Crear Investigador' button is located at the bottom right of the form.

Fuente: Autores del Proyecto

En la figura 40, se muestra la forma de administrador en la cual se crearan los usuarios del cuaderno electrónico, en el cual se requerirá la siguiente información: Nombre y apellido de investigador, Nivel de estudio, Áreas de conocimiento, Email, Identificación, Nombre de Usuario, Contraseña. En el momento de la creación del investigador se creara un archivo en un documento XML los usuarios verificando que no existan nombres de usuario repetidos y que posea los respectivos elementos que debe poseer un investigador nuevo.

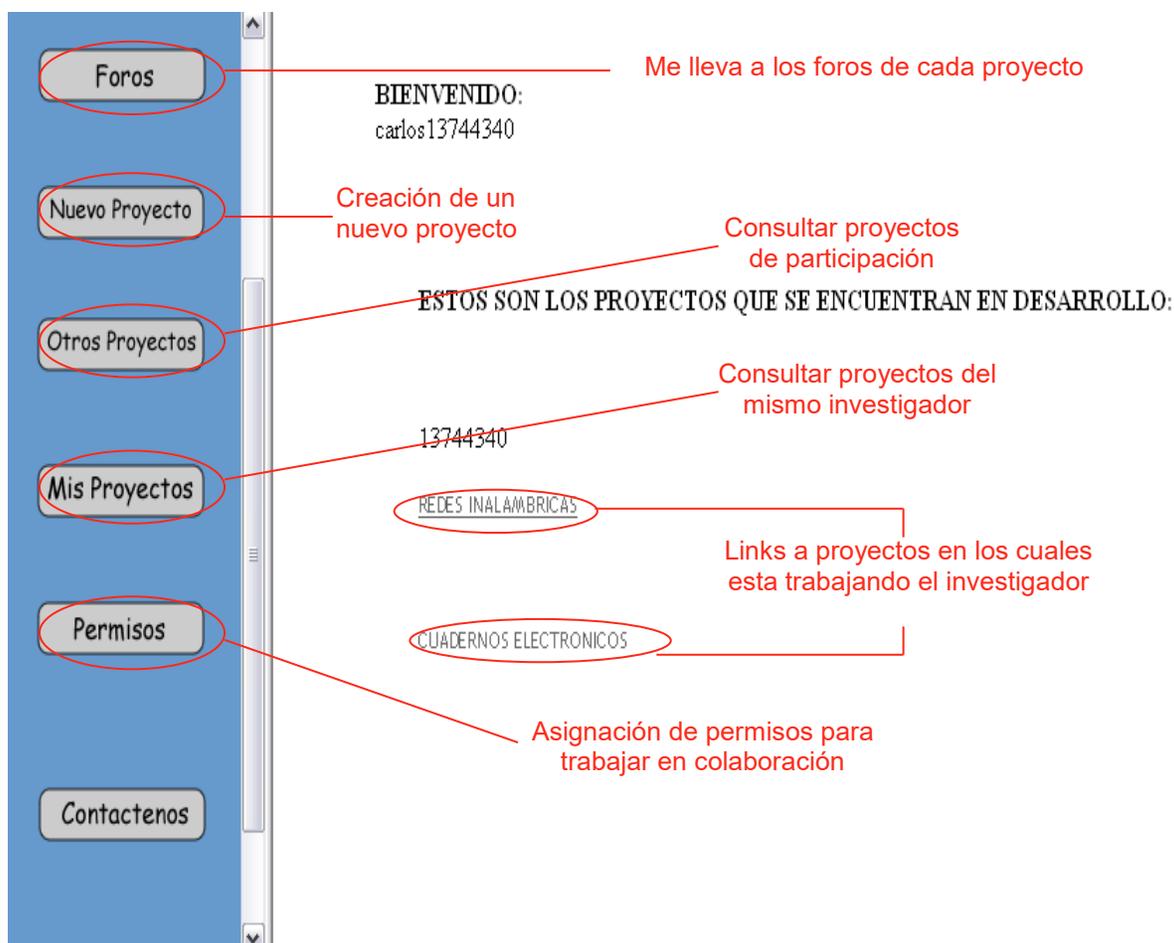
En la figura 41, se muestra como para poder trabajar en colaboración con otros investigadores, el usuario dueño de su proyecto concede los permisos escogiendo en una lista de usuarios con quienes desea trabajar junto con los proyectos a los cuales puede tener acceso solo para agregar foros y mirar anotaciones, sin la posibilidad de modificar o eliminar datos o proyectos. El sistema relaciona desde el inicio de sesión el identificador del investigador.

Figura 41. Gestión de permisos



Fuente: Autores del Proyecto

Figura 42. Proyectos en desarrollo

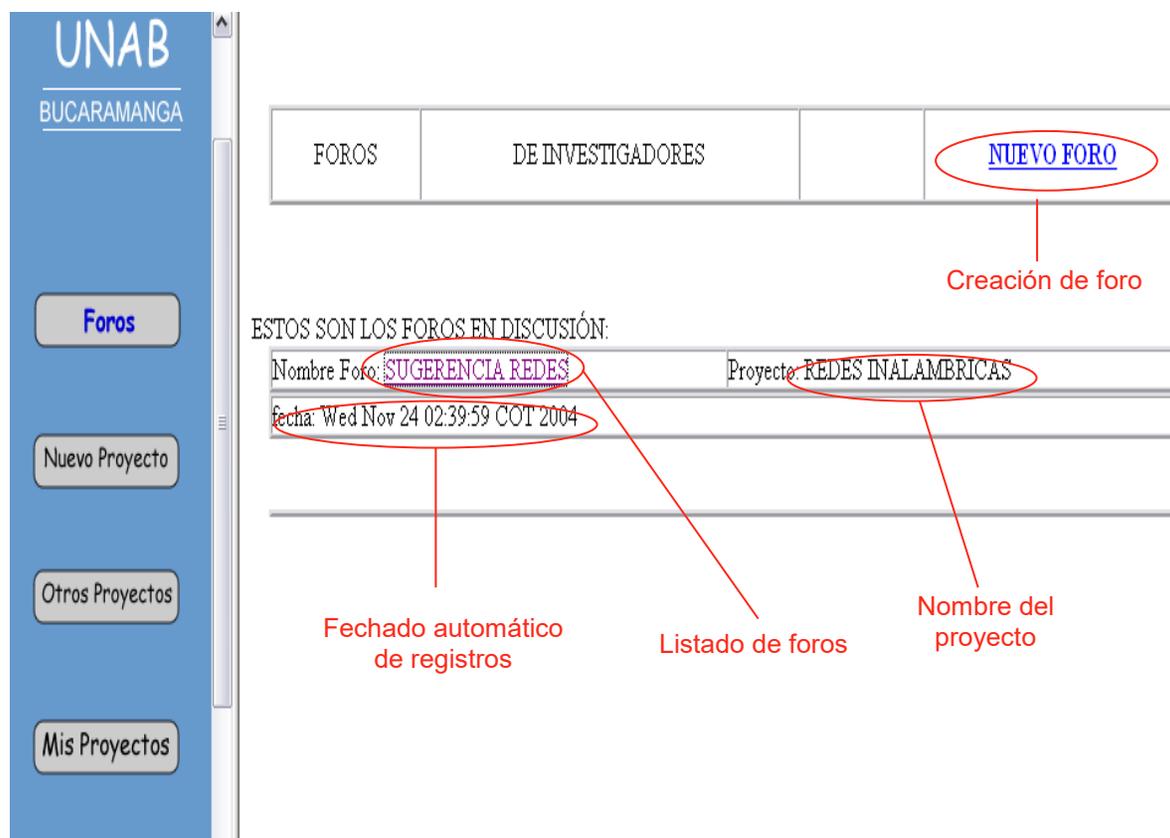


Fuente: Autores del Proyecto

En la figura 42, muestra los proyectos en los que actualmente está trabajando el investigador dueño del cuaderno y en los que esta colaborando con la investigación y desarrollo. También existe un link que lleva directamente a los foros hechos por los investigadores que están trabajando en colaboración en cada uno de los proyectos. La creación de un nuevo proyecto y la asignación de

permisos para trabajar en colaboración, también puede ser accedida desde esta página.

Figura 43. Creación de foros



Fuente: Autores del Proyecto

En la figura 43, se puede observar los foros que existen actualmente en cada uno de los proyectos. También se puede observar que sale automáticamente el fechado de cada anotación y también se puede crear un nuevo foro.

Figura 44. Creación de nuevos comentarios para foros

UNAB
BUCARAMANGA

Foros

Nuevo Proyecto

Otros Proyectos

Mis Proyectos

Permisos

NUEVO COMENTARIO DEL FORO

Nombre del foro: SUGERENCIA REDES

Título comentario:

Id Usuario: 13744340

Wed Nov 24 04:38:52 COT 2004

Crear comentario

Digitado por el investigador

Capturado desde inicio de sesión

Fechado automático

Crea un comentario para el foro

Fuente: Autores del Proyecto

En la figura 44 se puede observar como se puede hacer una nueva anotación a un foro. El investigador tiene que digitar el título y el contenido del comentario. Automáticamente se podrá ver el nombre al foro en el cual se está haciendo el comentario, el Identificador del usuario que se registra desde el inicio de sesión, al igual que el fechado

CONCLUSIONES

En conclusión, se han analizado las principales metodologías orientadas a agentes, y en particular la metodología INGENIAS, la cual ha sido estudiada y aplicada al DESARROLLO de Cuadernos electrónicos.

Las ventajas de enfocar el desarrollo multiagente se traduce en una mejora de la gestión, desarrollo, reutilización y mantenimiento de los sistemas. Esto se clarifica con la utilización, de los meta-modelos ofrecidos por INGENIAS. El meta-modelo de organización facilitó modelar la organización en la que se iba a introducir el sistema multiagente. El meta-modelo de agente permitió especificar los agentes que intervinieron en el sistema junto con sus responsabilidades y colaboraciones. El meta-modelo de objetivos y tareas permitió tener una visión del sistema basado en tareas y objetivos que debían ser cumplidos. El meta-modelo de entorno proporcionó una serie de actividades que permitieron modelar, organizar y estructurar la comunicación entre los agentes software. El meta-modelo de interacción permitió modelar la comunicación entre los agentes del sistema multiagente.

Dentro de la definición de la metodología, se realizó un análisis detallado, estructurado y profundo todos los meta-modelos. Se pudo ver claramente las ventajas que ofrece la metodología frente a las otras de manera que la diferenciación, entendimiento y manejo se hizo con más facilidad.

La desventaja que se encontró en la metodología INGENIAS era que se carecía de información y daba un esbozo muy superficial del meta-modelado. Por tales motivos se investigó en algunos ejemplos ofrecidos en la página de GRASIA y se decidió experimentar paso a paso el modelado, motivo por el cual se ocupó más tiempo de lo esperado.

La tecnología orientada a agentes es un tema que aún tiene múltiples campos en los cuales realizar investigación, el trabajo realizado en esta propuesta tocó aspectos importantes del paradigma orientado a agentes, pero no trató otros que pueden resultar en futuros proyectos de investigación.

REFERENCIAS BIBLIOGRAFICAS

BARRERA Sanabria Gareth, Rodríguez Buitrago Carolina. Aplicación de una metodología orientada a agentes en la implantación de un sistema de reserva de vuelos. Universidad Autónoma de Bucaramanga. Facultad de Ingeniería de Sistemas, 2001

BRAY, J. Paoli, C. M. Sperberg-McQueen, Nore Markup Language (XML) 1.0 (Second Edition). Disponible en: <http://www.w3.org/TR/REC-xml.html> W3C Recommendation 6 October 2000.

CASAS Castañeda Casas Castañeda Norma Judith, Quintanilla Diana Patricia. Tesis Diseño e implementación de un prototipo de comercio electrónico utilizando un paradigma orientado a agentes. Universidad Autónoma de Bucaramanga. Facultad de Ingeniería de Sistemas, 2002

CASTELFRANCHI, C. Guarantees for autonomy. En : Cognitive Agent Architecture. (1995). Citado por : IGLESIAS FERNÁNDEZ, Carlos Ángel. Definición de una metodología para el desarrollo de sistemas multiagente. España, 1998, 321 p. Tesis (Doctoral). Universidad Politécnica de Madrid. Departamento de Ingeniería de Sistemas Telemáticos.

CONNOLLY, F. van Harmelen, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. DAML+OIL Reference Description. W3C Note 18 December 2001. Disponible en <http://www.w3.org/TR/daml+oil-reference>.

DEAN, D. Connolly, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. OWL Web Ontology Language 1.0

Reference W3C Working Draft 29 July 2002. Disponible en <http://www.w3.org/TR/owl-ref>.

DECKER, S. Keith: *Environment Centered Analysis and Design of Coordination Mechanisms*. Informe. Department of Computer Science, University of Massachusetts. 1995

DIAZ Silva José Fabián, Murillo Anderson. Diseño e implementación de un prototipo de mercado virtual utilizando la tecnología de agentes. Universidad Autónoma de Bucaramanga. Facultad de Ingeniería de Sistemas, 2003.

ELLIOTTE Rusty Harold. XML Bible 3ra edición isbn: 076454986-3 February 2004

EURESCOM. MESSAGE: Methodology for engineering systems of software agents. Initial methodology. Technical Report P907-D1, EURESCOM. January 2000

EURESCOM. MESSAGE: Methodology for engineering systems of software agents (Final). Technical Report P907-TI1, EURESCOM. December 2001

FERBER, J. y Gutknecht, O.: A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems. Actas de conferencia. Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS98), IEEE CS Press. 1998.

GALLIERS, J. A theoretical framework for computer models of cooperative dialogue, acknowledging multiagent conflict. Tesis (PhD). Open University Uk. Citado por : IGLESIAS FERNÁNDEZ, Carlos Ángel. Definición de una metodología para el desarrollo de sistemas multiagente. 321 p. Tesis (Doctoral).

Universidad Politécnica de Madrid. Departamento de Ingeniería de Sistemas Telemáticos. España, julio 1998

GARCIA Juan Carlos. Buscadores inteligentes de información basados en la tecnología de agentes móviles. Universidad Autónoma de Bucaramanga. Facultad de Ingeniería de Sistemas, 2001.

GEIST, Al. Design of The DOE2000 Electronic Notebook : The Electronic Notebook Architecture. Berkeley California. December 1997

GILBERT, N. 1-85728-305-8, UCL Press, London, *Artificial Societies: the Computer Simulation of Social Life*. February 1995

GOMEZ, J. Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. No.18, pp. 51-63. ISSN: 1137-3601. © AEPIA (<http://www.aepia.dsic.upv.es/>). Metodologías para el desarrollo de Sistemas multi-agente Jorge J. Gómez Sanz 2003

GUTTMAN, R. H. y A. G. Moukas *The Knowledge Engineering Review*, Cambridge University Press, 0269-8889, Editado por Simons Parsons y Adele E. Howe, "Agent-mediated electronic commerce: a survey", , 1998, 147-159.

HYACINTH, Nwana. Software agents: An overview [online]. Disponible en: <http://labs.bt.com/proyects/agents/publish/papers/review.html> [cited 25 august 2004]

IBM Agent Building and Learning Environment (ABLE). [online] Available from World Wide Web: <<http://www.alphaworks.ibm.com/tech/able>> [cited 15 february 2004]

IGLESIAS Fernández, Carlos Ángel *et al.* Analysis and design of multiagent systems using MAS – CommonKADS. Valladolid, España. (1999); 15 p.

JENNINGS, J. International Journal of Cooperative Information Systems, World Scientific Publishing Co., 0218-8430, Editado por M. P. Papazoglou, "Agent-based business process management.", N. R. April 1996.

D'INVERNO, Mark y Michael Luck, 3-540-41975-6, Springer, *Understanding Agent Systems*. March 2001

JACOBSON, I., Booch, G. y Rumbaugh, J.: El Proceso Unificado de Desarrollo de Software. Libro completo. Addison Wesley. 303-3792000. January 2000

MALONE, T. W. and Crowston, K., *The Interdisciplinary Study of Coordination*, ACM Computing Survey, vol. 26, no. 1, pp. 87-119, Mar.1994

MARTINEZ Eduardo, Prieto William y Freddy Pico. Prototipo de aplicación de comercio electrónico utilizando la metodología Gaia al desarrollo de software orientado a agentes. Universidad Autónoma de Bucaramanga. Facultad de Ingeniería de Sistemas, 2002.

MATURANA, H.:Autopoiesis, Structural Coupling and Cognition. International Society for the System(s) Science(s) (ISSS). [online] Available from World Wide Web: <http://www.iss.org/maturana.htm> [cited 1 march 2004]

MYERS, James D., Elena S. Mendoza , Bonnie Hoopes. A collaborative electronic laboratory notebook , Pacific Northwest, National Laboratory PO Box 999 Richland, WA 99352 USA

MONTAGÚ Castro, María Clemencia, Vargas Mayorga, Jorge Leonardo. Tesis Aplicación de la metodología ingenierías en la implementación de un prototipo de Supply Chain Manageme. Universidad Autónoma de Bucaramanga. Facultad de Ingeniería de Sistemas, 2004

NEWELL, A. and Simons, H. A., *GPS: A program that simulates Human Thought, en Computers and Thought*. Mc Graw Hill, 1963.

O. LASSILLA, O., R. R. Swick. Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation 22 February 1999. Available at <http://www.w3.org/TR/REC-rdf-syntax>.

PERRY, Bruce W. Java Servlet & JSP Cookbook . Publisher : O'Reilly Pub Date : January 2004 ISBN : 0-596-00572-5 Pages : 746

RAO, A y M. Georgeff. Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference, Morgan Kaufmann, San Mateo, 1-55860-262-3, Editado por B. Nebel, C. Rich, y W. Swartout, "An abstract architecture for rational agents, 1992.

RICH, E. y Knight, K.: *Artificial Intelligence*. Libro completo. McGraw-Hill. 1990

ROSENSCHEIN, J and GENESERETH, M. Deals among rational agents. En : Proceedings of the ninth International joint conference on artificial intelligent. (1985)

RUSELL, S. y Norvig, P: *Artificial Intelligence: a modern approach*. Libro completo. Prentice Hall. 1995.

SYCARA, K., Klusch, M., idof, S., and Lu, J., *Dynamic Service Matchmaking among Agents in Open Information Environments*, Journal ACM SIGMOD Record, Special Issue on Semantic Interoperability in Global Information Systems, 1999

TIMBERNERS-LEE, J. Hendler, O Lassila. Fascinating facts about Tim Berners-Lee inventor of the World Wide Web. The Semantic Web 12-589-6587-AK25, <http://www.ideafinder.com/history/inventors/berners-lee.htm> Scientific American, May 2001

TVEIT, AMUND; A survey of Agent-Oriented Software Engineering, Norwegian University of Science and Technology, May 2001.

WAGNER G. Agent-Oriented Analysis and Design of Organizational Information Systems. In *Proc. of Fourth IEEE International Baltic Workshop on Databases and Information Systems, Vilnius (Lithuania)*, May 2000.

Wim Coulier: Belgacom Project Leader & Responsible for Dissemination. Disponible en: <http://www.eurescom.de/~public-webSPACE/P900-series/P907> May 23, 2000

WHITAKER, R.: *Self-Organization, Autopoiesis, and Enterprises*. ACM SIGGROUP. <http://www.acm.org/siggroup/auto/Main.html>

WOOLDRIDGE, Michael et al. 0-471-49691-X, Agent – oriented software engineering for Internet applications. An introduction to Multiagent Systems October 2002.

WOOLDRIDGE and N. R. Jennings. Agent theories, architectures, and languages: A survey. In M. Wooldridge and N. R. Jennings, editors, *Intelligent Agents*:

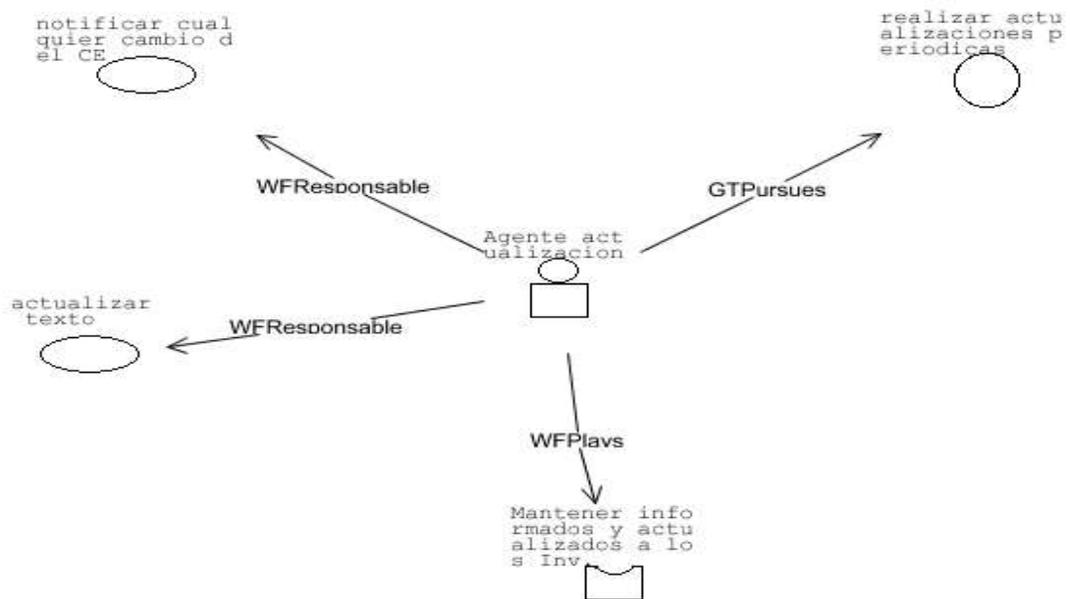
Theories, Architectures, and Languages (LNAI Volume 890), Springer-Verlag: Heidelberg, Germany, Jan. 1995.

Workflow Management Coalition: *The Workflow Management Coalition Specification: Workflow Management Coalition Terminology & Glossary*. Informe. 1999

ANEXO 1

METAMODELO DE AGENTES

Figura 21. Agente Actualización

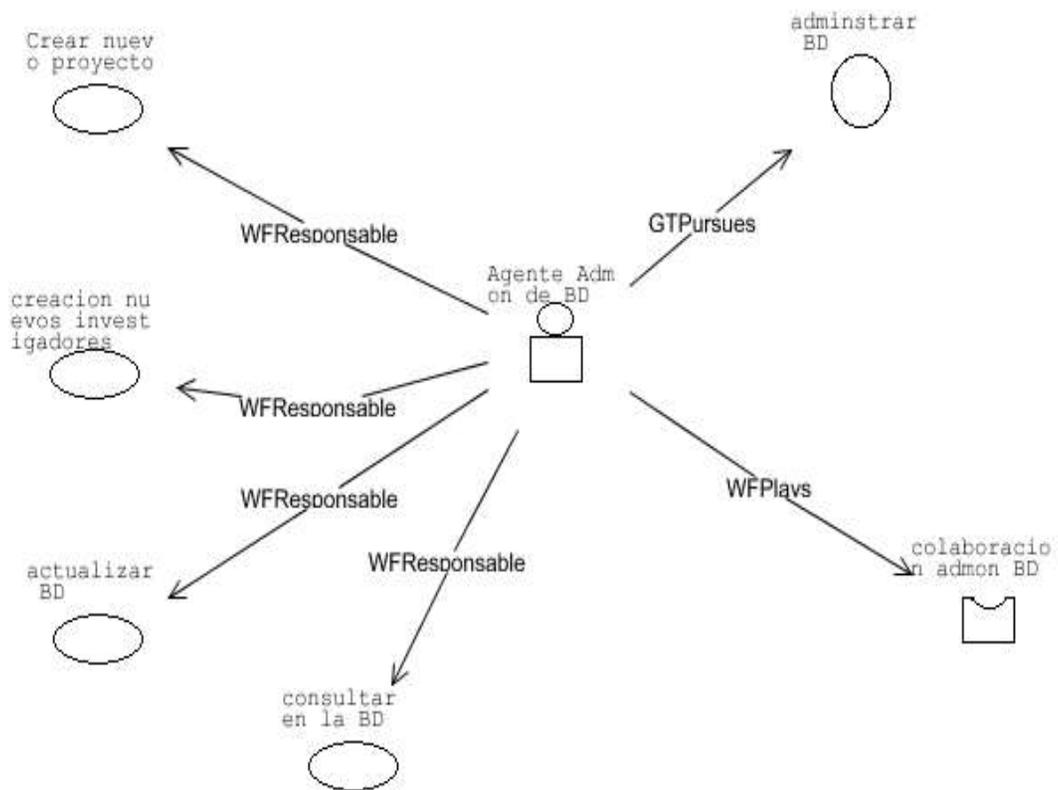


Fuente: Autores del Proyecto

ANEXO 2

METAMODELO DE AGENTES

Figura 22. Agente Administración de Base de Datos

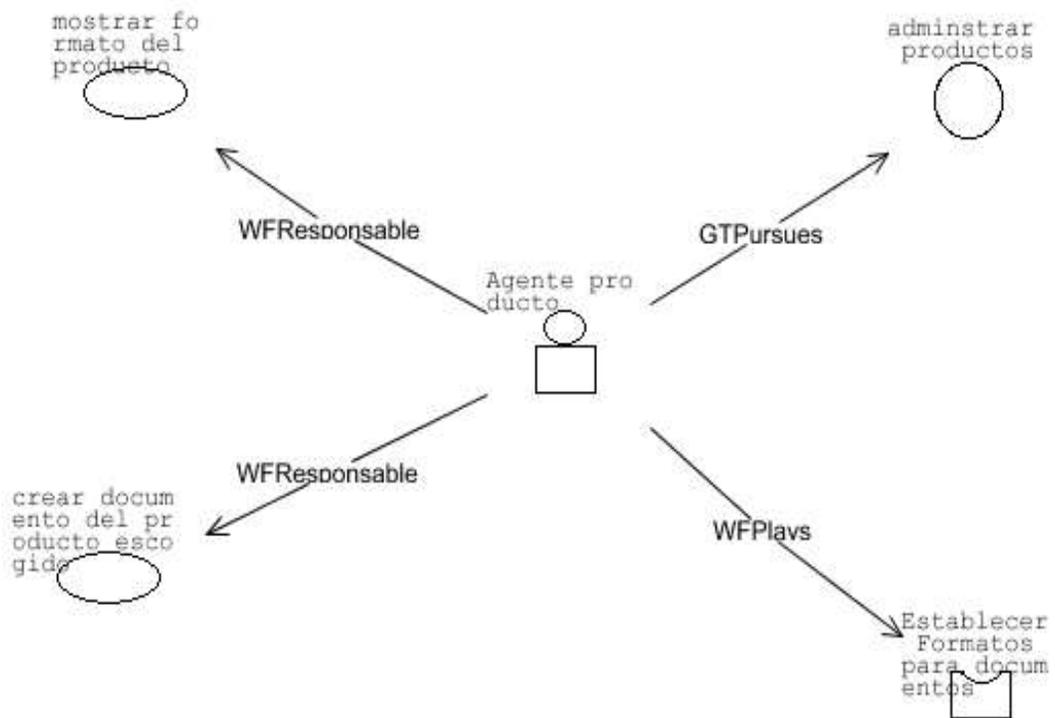


Fuente: Autores del Proyecto

ANEXO 3

METAMODELO DE AGENTES

Figura 23. Agente Producto

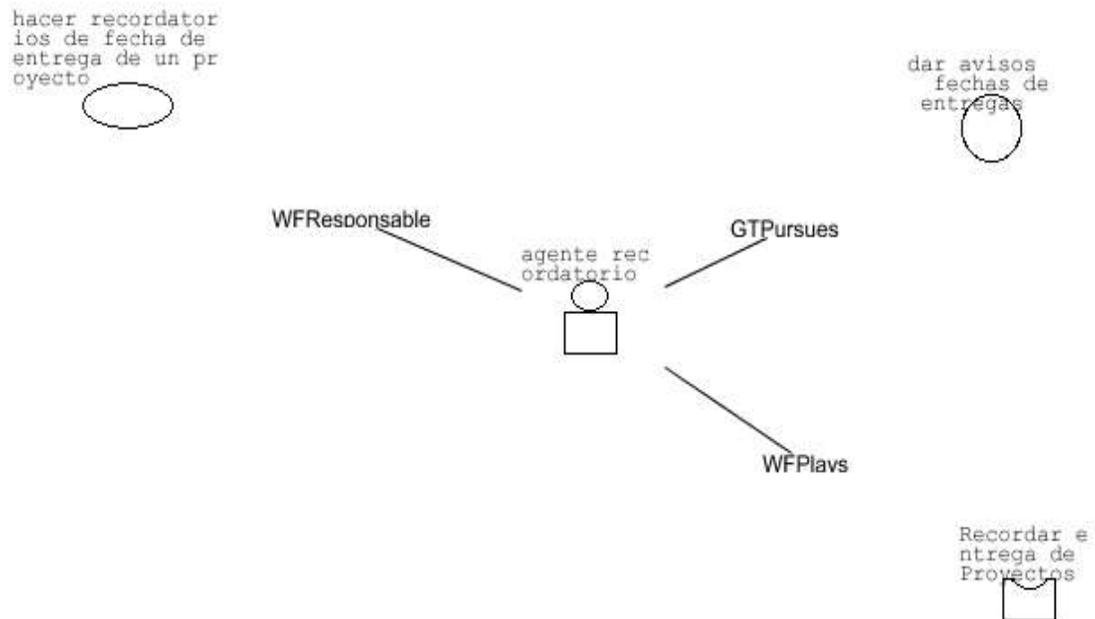


Fuente: Autores del Proyecto

ANEXO 4

METAMODELO DE AGENTES

Figura 24. Agente Recordatorio

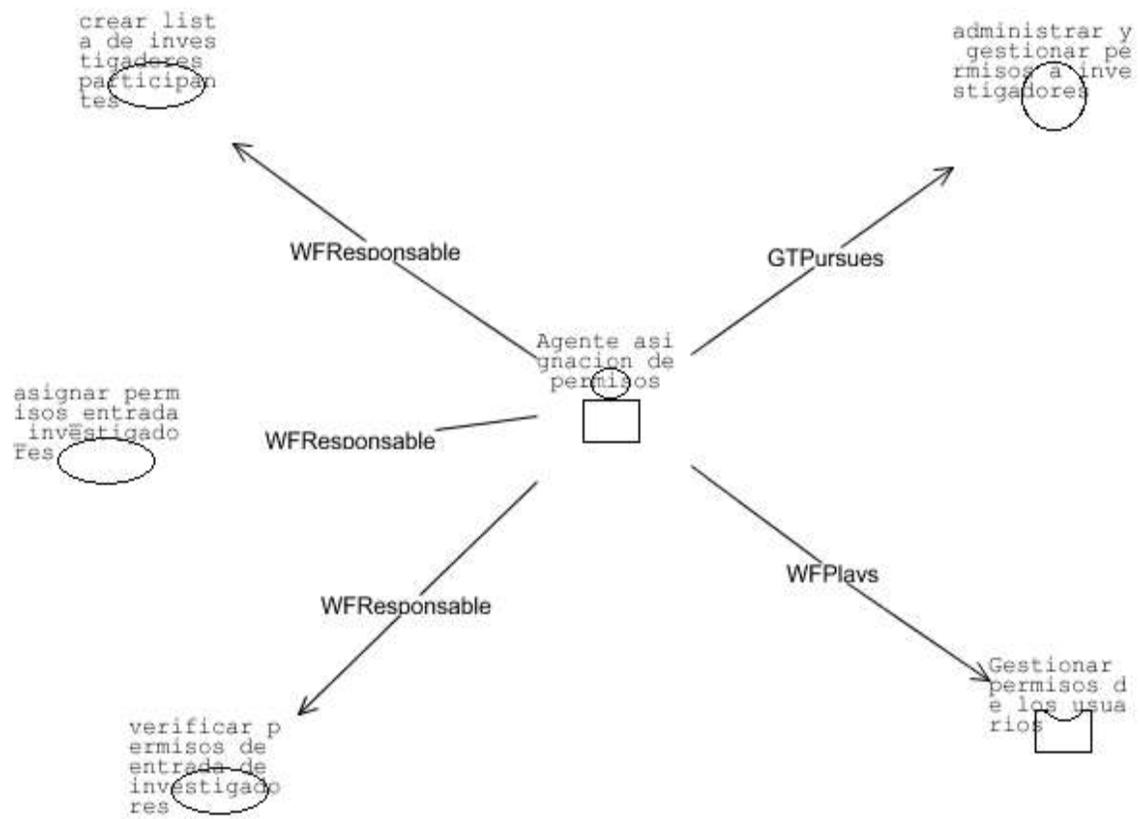


Fuente: Autores del Proyecto

ANEXO 5

METAMODELO DE AGENTES

Figura 25. Agente Asignación de Permisos

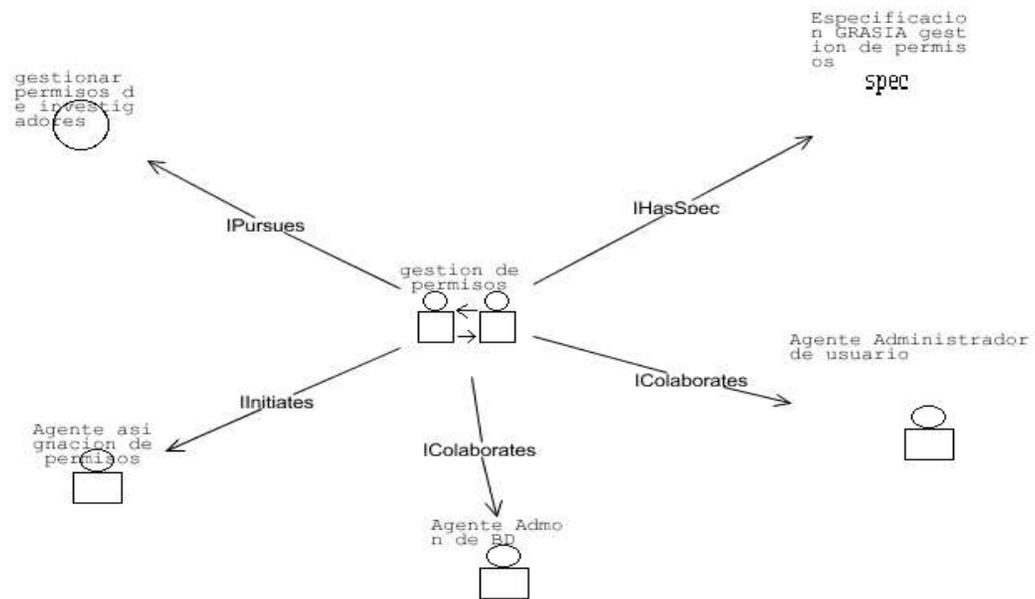


Fuente: Autores del Proyecto

ANEXO 6

META-MODELO DE INTERACCION

Figura 29. Gestión de Permisos

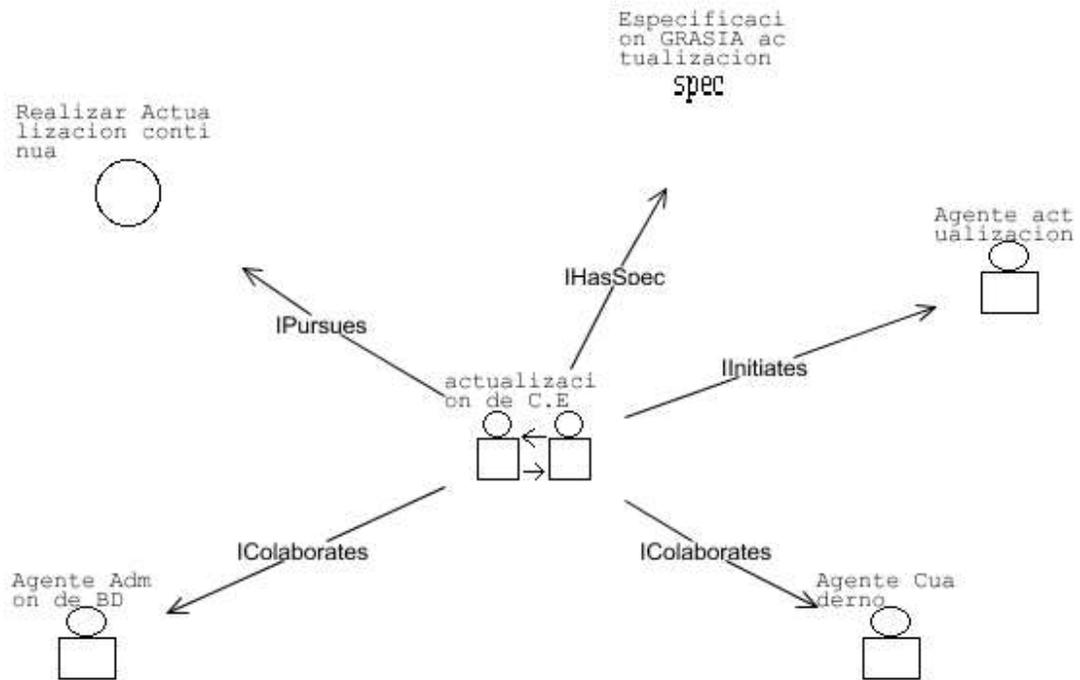


Fuente: Autores del Proyecto

ANEXO 7

META-MODELO DE INTERACCION

Figura 30. Actualización del Cuaderno Electrónico

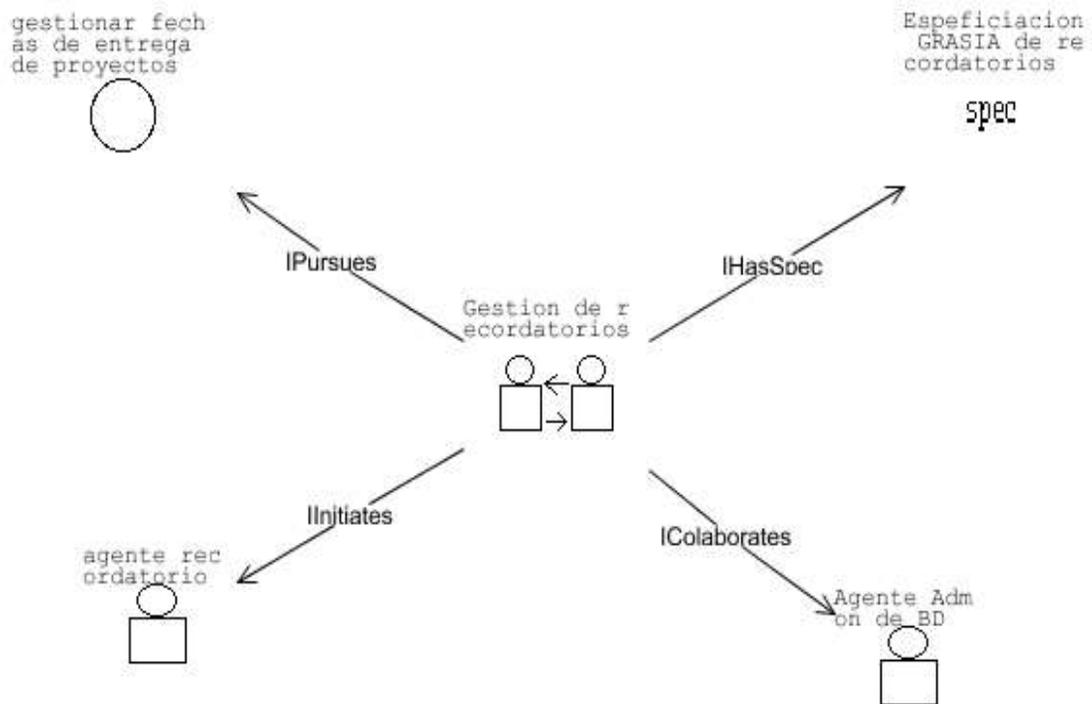


Fuente: Autores del Proyecto

ANEXO 8

META-MODELO DE INTERACCION

Figura 31. Gestión de Recordatorios

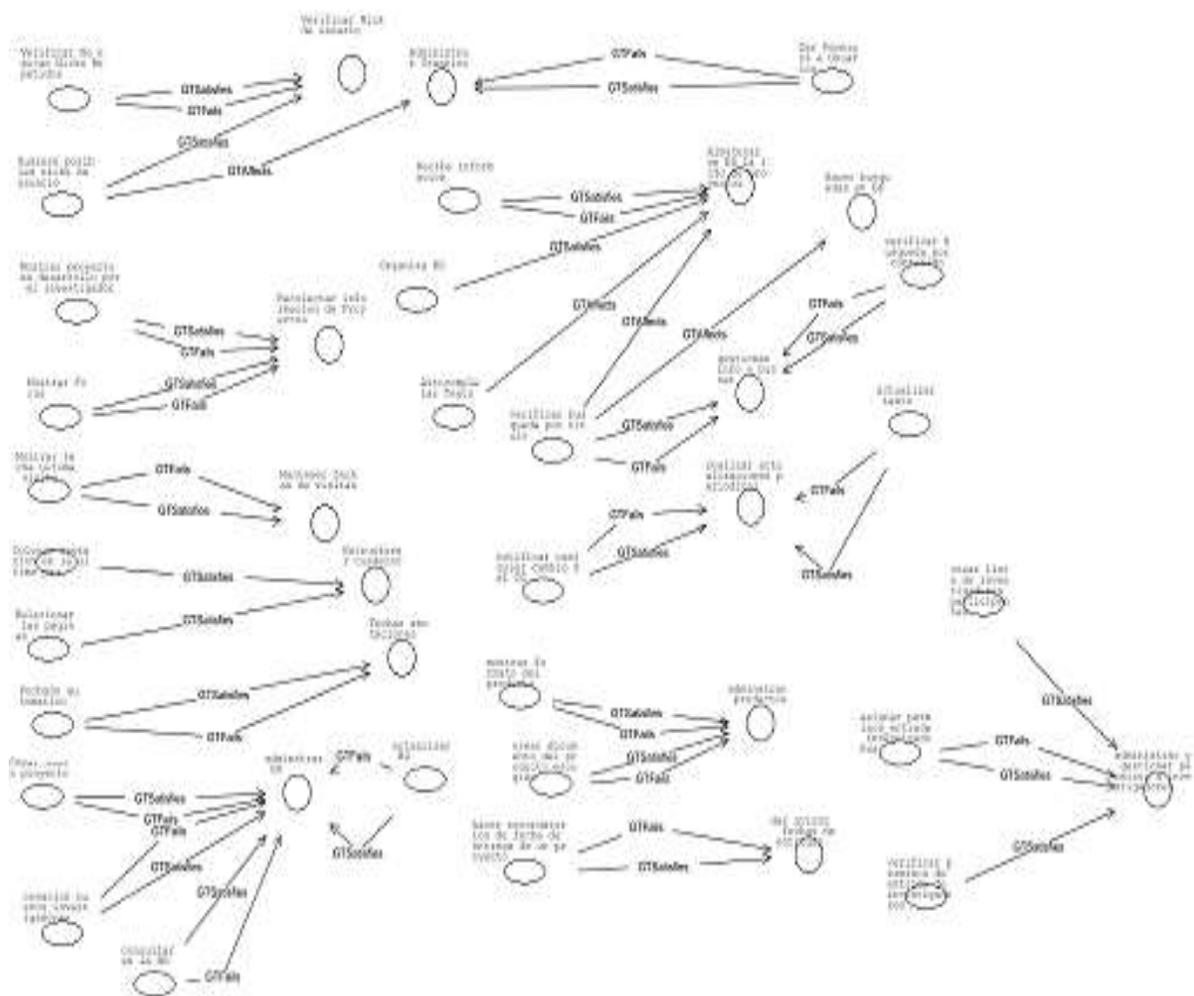


Fuente: Autores del Proyecto

ANEXO 9

META-MODELO DE OBJETIVOS Y TAREAS

Figura 33. Satisfacción de objetivos

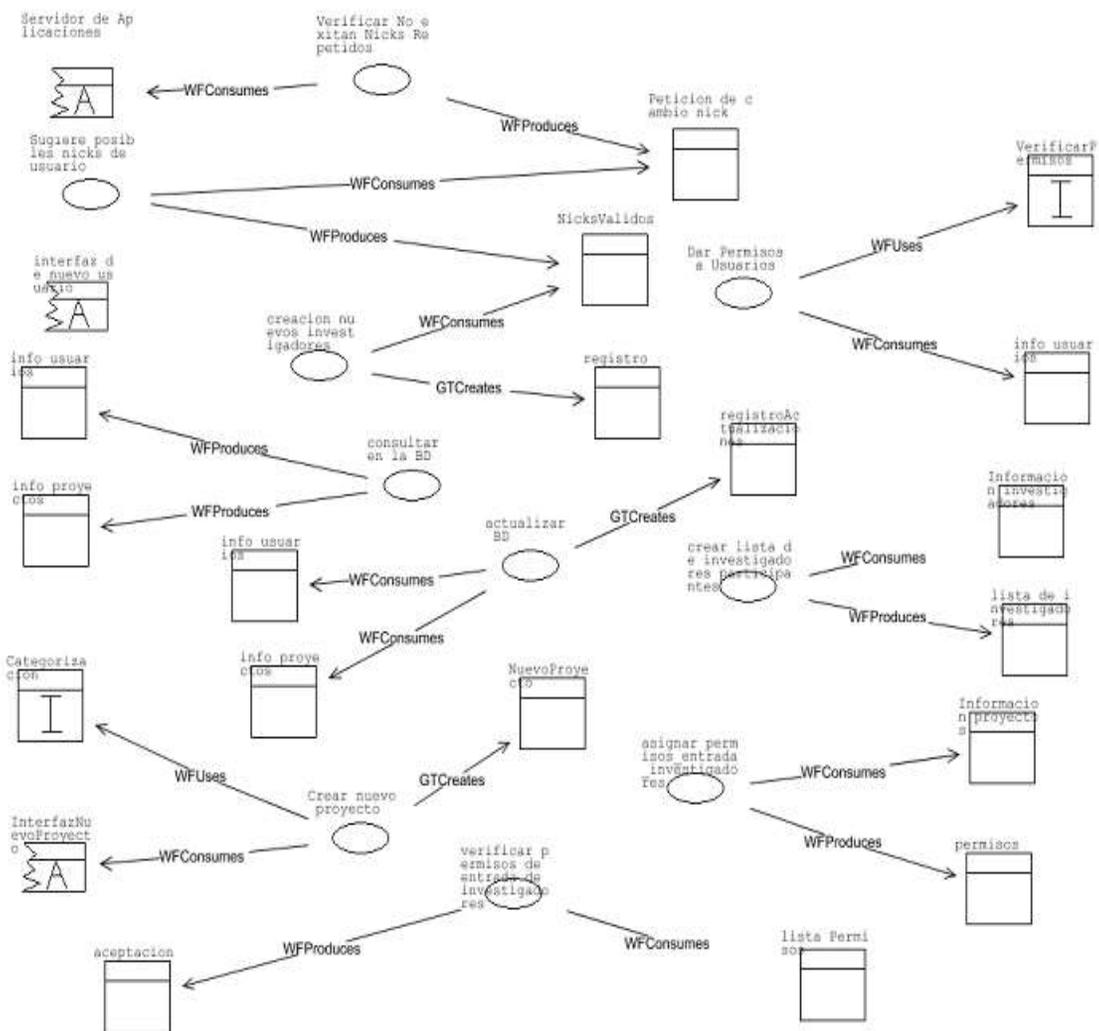


Fuente: Autores del Proyecto

ANEXO 10

META-MODELO DE OBJETIVOS Y TAREAS

Figura 34. Flujo de Tareas

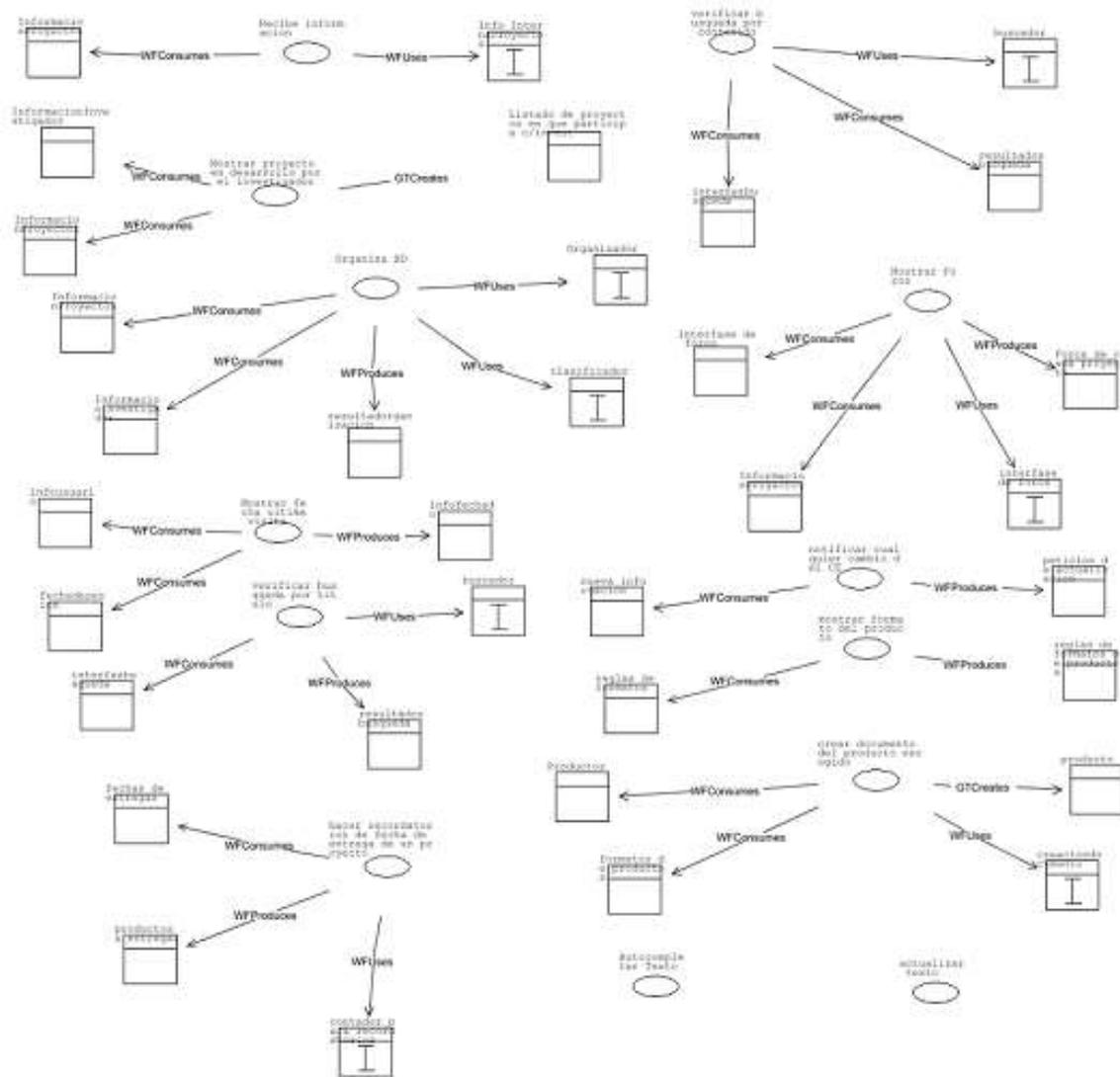


Fuente: Autores del Proyecto

ANEXO 11

META-MODELO DE OBJETIVOS Y TAREAS

Figura 35. Flujo de Tareas específico



Fuente: Autores del Proyecto