

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE INGENIERÍA

AGENTE INTELIGENTE DE RECOMENDACIÓN DE PRODUCTOS O SERVICIOS BASADO EN ONTOLOGÍAS

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

PATRICIO ANTONIO CHIRIBOGA CORONEL

DAVID RENATO RACINES MEDINA

DIRECTOR: ING. DIEGO SAA

Quito, julio de 2006

DECLARACIÓN

Nosotros, Patricio Antonio Chiriboga Coronel y David Renato Racines Medina, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Patricio Chiriboga

David Racines

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Patricio Antonio Chiriboga Coronel y David Renato Racines Medina, bajo mi supervisión.

Ing. Diego Saa

DIRECTOR DE PROYECTO

CONTENIDO

CAPÍTULO 1.	Marco teórico	7
1.1.	Comportamiento del consumidor y marketing sobre Internet.....	8
1.1.1.	Aspectos generales sobre e-marketing.....	8
1.1.2.	Estrategia de e-marketing.....	10
1.1.3.	Comunicación para e-marketing	17
1.1.4.	Modelo del comportamiento del consumidor	22
1.2.	Ontologías.....	27
1.2.1.	Aspectos generales	27
1.2.2.	Criterios de diseño y proceso de construcción de ontologías	32
1.2.3.	Lenguajes de representación.....	35
1.2.4.	Aplicaciones en el ámbito del e-business	37
1.3.	Agentes inteligentes.....	39
1.3.1.	Aspectos generales	39
1.3.2.	Tipos de agentes	43
1.3.3.	Agentes como mediadores en el e-commerce.....	45
1.3.4.	Agentes de recomendación de productos.....	47
CAPÍTULO 2.	Análisis y diseño del Agente.....	52
2.1.	Metodología de desarrollo.....	52
2.2.	Análisis.....	52
2.2.1.	Contexto del problema	52
2.2.2.	Descripción del problema	52
2.2.3.	Alcance	53
2.2.4.	Visión general de la aplicación.....	54
2.2.5.	Especificación de requerimientos	55
2.2.6.	Modelo de casos de uso	55
2.2.7.	Arquitectura del software a nivel de análisis	59
2.3.	Diseño.....	61
2.3.1.	Arquitectura del software a nivel de diseño	61
CAPÍTULO 3.	Implementación y pruebas del Agente.....	77
3.1.	Implementación.....	77
3.1.1.	Selección de herramientas de implementación	77
3.1.2.	Descripción de herramientas y plataforma de desarrollo	84

3.1.3.	Selección de versiones de herramientas	92
3.1.4.	Modelo de implementación	94
3.1.5.	Detalles de implementación	99
3.2.	Caso de estudio	118
3.3.	Pruebas.....	119
3.3.1.	Modelo de pruebas para el Agente de Recomendación	120
CAPÍTULO 4.	Conclusiones y recomendaciones	124
4.1.	Conclusiones	124
4.2.	Recomendaciones	125
	Referencias bibliográficas	128
	Acrónimos	132
	Glosario.....	134

RESUMEN

El presente proyecto de titulación consiste en el desarrollo de un agente de recomendación de productos o servicios que utiliza técnicas de inteligencia artificial. Dicho agente está destinado a trabajar en ambientes de comercio electrónico sobre la plataforma de Internet; además, es capaz de soportar las fases iniciales de procesos de compra al ayudar a los clientes a reconocer necesidades y presentar alternativas de información.

Se definen varios aspectos de recomendación relacionados con partes y accesorios, categorías de productos y perfiles de usuarios. Por otro lado, se extraen varios elementos de recomendación que proceden de una estrategia de marketing como productos y servicios. Adicionalmente, se consideran promociones y publicidad ya que son elementos que involucran directa o indirectamente productos y servicios. Los requerimientos funcionales del Agente de Recomendación surgen a partir de la combinación de los aspectos y elementos de recomendación especificados.

La metodología de desarrollo de software utilizada en el proyecto es el Rational Unified Process (RUP) con ciertas particularidades orientadas a la tecnología J2EE. En la fase de análisis se determinan los casos de uso los cuales están bien estructurados y clasificados por categorías; lo que permite la creación de una arquitectura de software híbrida que combina conceptos de sistemas orientado a objetos, sistemas basados en conocimiento y agentes inteligentes.

El producto resultante de la fase de implementación es un servicio Web listo para ser utilizado exclusivamente por otros sistemas como por ejemplo agentes de software y aplicaciones de e-commerce.

Por otra parte, al terminar la fase de pruebas se determina que la arquitectura de software planteada es totalmente efectiva pero poco eficiente; sin embargo las conclusiones y recomendaciones obtenidas al final plantean inquietudes interesantes y conforman la base para trabajos futuros.

PRESENTACIÓN

El Internet ha revolucionado la forma tradicional de realizar actividades comerciales y de marketing. Por tal motivo, las empresas han empezado a buscar modelos de negocios cada vez más competitivos mediante la creación y diversificación de nuevas alternativas de comunicación.

Una estrategia de marketing completa incluye un análisis detallado del mercado; el marketing provee los elementos básicos para orientar la decisión de compra a través del estudio del comportamiento del consumidor, en donde se describe la manera de actuar del comprador y el entorno en el que se desenvuelve.

Por otra parte, el intercambio interactivo de información sobre la red presta múltiples ventajas para la promoción y distribución personalizada de productos y servicios. De esta manera la empresa fortalece la relación con sus clientes lo que incrementa la calidad de servicio y garantiza la lealtad del consumidor.

La tecnología de agentes inteligentes es producto de la síntesis de todas las disciplinas de la inteligencia artificial. La aplicación de esta tecnología en el campo de los negocios electrónicos automatiza un gran número de tareas que necesitan de la intervención humana. Por esta razón, los agentes que soportan los procesos de e-commerce necesitan incluir el conocimiento de expertos como compradores, vendedores y especialistas de marketing. De esta manera se pueden conseguir de una forma más efectiva los objetivos de un negocio electrónico y las actividades comerciales sobre Internet pueden contar con mejores servicios personalizados y eficientes.

Por otra parte, en el contexto de la filosofía se define ontología como la descripción de lo que existe. Históricamente, este término ha tenido diferentes significados; sin embargo, los enfoques filosóficos contemporáneos utilizan herramientas matemáticas como la lógica formal para analizar las categorías fundamentales de objetos y estudios de las relaciones de dependencia entre la parte y el todo.

El campo de las ciencias de la computación ha adoptado este tópico para estudiarlo desde una nueva perspectiva que contempla cuestiones relacionadas con representación del conocimiento. Además, es necesario señalar que las ontologías se han convertido, en los últimos años, en el foco de investigación de diversas comunidades científicas debido a su gran potencial de aplicación a

través de las tecnologías de la información. Los principales beneficios se basan en la simplificación de tareas relacionadas con la representación, intercambio y reutilización del conocimiento. De esta manera se da un nuevo matiz a todas las posibilidades de comunicación entre los sistemas de información y sus usuarios al mejorar su capacidad de interacción.

CAPÍTULO 1. MARCO TEÓRICO

1.1. COMPORTAMIENTO DEL CONSUMIDOR Y MARKETING SOBRE INTERNET

El Internet ha revolucionado la forma tradicional de realizar actividades comerciales y de marketing. Por tal motivo, las empresas han empezado a buscar modelos de negocios cada vez más competitivos mediante la creación y diversificación de nuevas alternativas de comunicación.

Una estrategia de marketing completa incluye un análisis detallado del mercado. El marketing provee los elementos básicos para orientar la decisión de compra a través del estudio del comportamiento del consumidor, en donde se describe la manera de actuar del comprador y el entorno en el que se desenvuelve.

Por otra parte, el intercambio interactivo de información sobre la red presta múltiples ventajas para la promoción y distribución personalizada de productos y servicios. De esta manera la empresa fortalece la relación con sus clientes lo que incrementa la calidad de servicio y garantiza la lealtad del consumidor.

1.1.1. ASPECTOS GENERALES SOBRE E-MARKETING

Las nuevas tecnologías de comunicación aplicadas al marketing tradicional dan lugar a una nueva generación de mercadeo. E-marketing es un proceso de administración que genera relaciones de intercambio satisfactorias, basadas en tecnologías de comunicación electrónica a través de la creación, promoción, distribución y venta de productos, servicios e ideas.

Dentro de la economía digital algunos términos pueden causar confusión debido a que su contexto esta íntimamente relacionado. Por esta razón es necesario comprender las siguientes definiciones (Chaffey 2005):

- **Internet marketing:** Es una aplicación del Internet, tecnologías digitales y la comunicación tradicional para lograr los objetivos del marketing.
- **E-commerce:** Es el conjunto de actividades financieras e intercambio de información, entre una organización y sus clientes a través de medios electrónicos.
- **E-business:** Es el conjunto de actividades financieras e intercambio de

información entre diferentes organizaciones a través de medios electrónicos que soportan un rango de procesos del negocio.

1.1.1.1. Modelo de transacciones entre el consumidor y el negocio

Una organización realiza transacciones con sus clientes u otras organizaciones, razón por la cual se presentan las siguientes posibilidades (Figura 1.1):

	Business	Consumer
Business	B2B	B2C
Consumer	C2B	C2C

Figura 1.1 Modelo de transacciones entre el consumidor y el negocio (Chaffey 2005).

- **Business-to-consumer (B2C):** Es el conjunto de transacciones comerciales entre una organización y sus clientes.
- **Business-to-business (B2B):** Es el conjunto de transacciones comerciales entre diferentes organizaciones.
- **Consumer-to-consumer (C2C):** Es el conjunto de transacciones comerciales entre clientes, a través de un portal de negocio.
- **Consumer-to-business (C2B):** Es el conjunto de transacciones comerciales en donde el cliente propone una oferta al negocio.

1.1.1.2. Beneficios que Internet provee al marketing.

La tecnología digital permite una comunicación directa con el cliente. Además incrementa el grado de personalización de las ofertas de productos y servicios mientras que se reducen los costos publicitarios. El nivel de satisfacción aumenta a través de la particularización y sofisticación de los procesos de marketing. Por lo tanto, las ventajas entre el cliente y la empresa son mayores. Los principales beneficios se detallan a continuación:

- Un cliente puede acceder a un sitio electrónico desde cualquier lugar del mundo.
- Los negocios por Internet operan las 24 horas del día durante todo el año.
- Reducción de costos publicitarios.
- Valor agregado a los productos a través de la información.
- Los costos de mantenimiento de inventarios físicos se reducen drásticamente.
- Aumento del número de clientes.
- La información de los productos se distribuye en forma inmediata.

- La abundancia de información permite al cliente comparar y escoger los productos minuciosamente.

1.1.2. ESTRATEGIA DE E-MARKETING

Una estrategia es un plan de acción diseñado para alcanzar las metas a largo plazo (Dalrymple 1990). Un plan de marketing ayuda a alcanzar una parte de los objetivos planteados por una organización. En el contexto del e-marketing, este plan debe utilizar Internet como medio de comunicación fundamental además de otras



Figura 1.2 Entorno del marketing (Los Autores).

tecnologías. Los elementos que intervienen en la estrategia de e-marketing son (Figura 1.2): Mercado meta, marketing mix y entorno del marketing.

En base a la segmentación un mercado global se escoge el mercado meta el cual es el centro de la estrategia. En este punto es donde interviene el proceso de marketing mix al combinar diferentes alternativas de producto, precio, distribución y promoción. El gerente de marketing se encarga de la planificación, control y evaluación de esta estrategia, la cual debe ir acompañada del análisis de varios factores de entorno como por ejemplo: el entorno económico, sociocultural e incluso tecnológico.

1.1.2.1. Mercado meta

Un cliente satisfecho es la clave del éxito de todo negocio. Este es el motivo por el cual los mercados de consumidores constituyen la razón de ser de una organización. Para generar un buen nivel de satisfacción en el cliente es necesario hacer un minucioso análisis del consumidor enfocado a los diferentes tipos de necesidades, deseos y requerimientos. Por esta razón, algunas compañías adquieren una mejor posición al servir a uno u otro segmento de mercado a través de la aplicación de diferentes estrategias.

Las principales actividades para la selección y orientación hacia el mercado meta son: análisis del consumidor, segmentación de mercado, diferenciación y

posicionamiento del producto.

1.1.2.1.1. Análisis del consumidor

Al estudiar al cliente con diferentes enfoques se pueden crear estrategias de marketing que satisfagan sus necesidades de una manera exitosa. El proceso de análisis del consumidor está basado en los siguientes aspectos:

- Tipos de consumidores
 - Localización de clientes
 - Consumo en relación al tiempo
 - Necesidades de los clientes
 - Comportamiento del consumidor
- **Tipos de consumidores:** Para identificar mejor las necesidades del mercado es imprescindible diferenciar los grupos de clientes existentes. La clasificación de clientes está en función de diferentes enfoques; sin embargo existen dos categorías fundamentales a las que el marketing les da un tratamiento especial: los clientes finales y los clientes industriales. Los clientes finales son aquellos que compran productos para su propio consumo; mientras que los clientes industriales son empresas que consumen cierto tipo de recurso o cumplen el papel de intermediarios de ventas.
 - **Localización de clientes:** Saber dónde están localizados los clientes es primordial para la estrategia de e-marketing. De esta forma se puede hacer un plan de comunicación orientado al mercado meta. La manera más común de localizar clientes es a través de una distribución geográfica.
 - **Consumo en relación al tiempo:** Permite identificar las temporadas en las que se compra más. Con esta información se puede modificar los patrones de compra para incentivar con promociones en las épocas de temporada baja.
 - **Necesidades de los clientes:** Para saber qué vender es necesario determinar las necesidades, deseos o requerimientos de los consumidores. Esta información se la puede obtener a través de estudios de tendencias de compra. El marketing estadístico ofrece una gran variedad de procedimientos para obtener este tipo de información.
 - **Comportamiento del consumidor:** Uno de los estudios más complejos dentro del análisis del consumidor es determinar por qué los clientes adquieren productos. Este es un proceso en donde intervienen un sin número de factores; sin embargo, para simplificar su estudio se lo sintetiza en modelos

de comportamiento del consumidor los cuales ayudan a comprender mejor la conducta del cliente (Ver sección 1.1.4).

1.1.2.1.2. Segmentación de mercado

La segmentación es un proceso por medio del cual se divide un mercado total en grupos de clientes relativamente homogéneos (Thomson 2005). Una estrategia de marketing se enfoca con más eficacia mediante la división del mercado en varios segmentos con características comunes. Para hacer una segmentación de mercado es necesario tomar en cuenta tres criterios fundamentales:

- Un segmento debe ser un grupo que permita hacer estudios estadísticos.
- Un segmento debe ser accesible para la comunicación.
- Un segmento debe ser lo suficientemente grande como para ofrecer un beneficio potencial.

Existen varios métodos de segmentación de mercado los cuales principalmente van en función de parámetros demográficos y geográficos. La Figura 1.3 muestra los parámetros de segmentación de mercado para los dos tipos de clientes: finales e industriales.

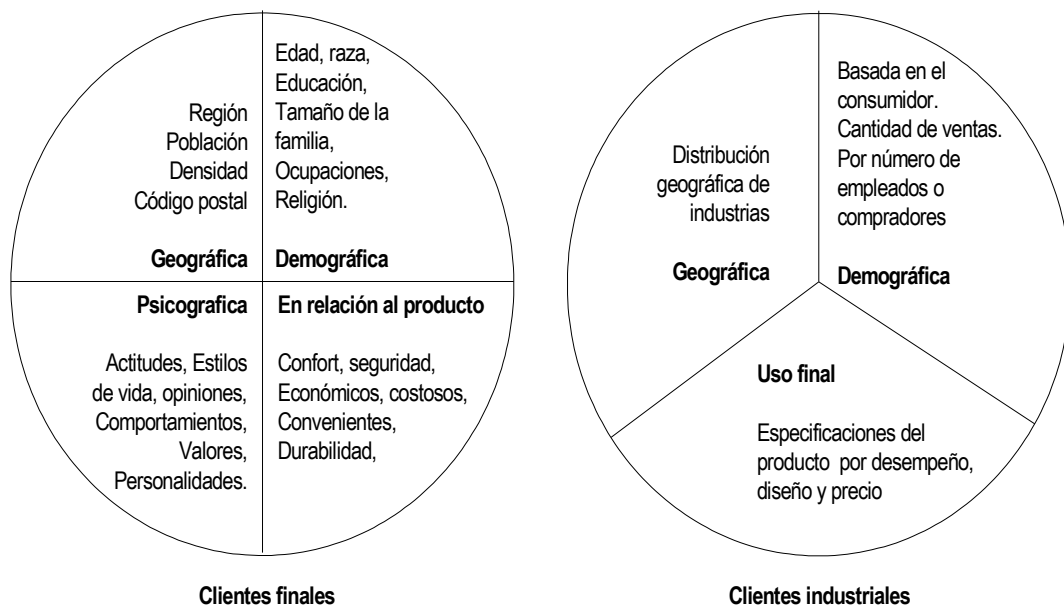


Figura 1.3 Parámetros de segmentación de mercado de acuerdo al tipo de cliente (Thomson 2005)

1.1.2.1.3. Diferenciación y posicionamiento del producto

La segmentación de mercado no tiene ningún valor si no se hace una diferenciación de productos. El proceso de diferenciación consiste en dar cierto

tipo de características especiales a los productos o servicios con el objetivo de atraer a un segmento de mercado. La combinación de una estrategia de segmentación con una estrategia de diferenciación depende del tipo de bien de consumo que se promocioe y los segmentos de mercados a los que están orientados los productos.

Después de la diferenciación, es necesario hacer un posicionamiento de los bienes de consumo. Posicionar en el mercado es disponer que un producto ocupe un lugar claro, distintivo y deseable en la mente de los consumidores meta, en relación con los productos de la competencia (Kotler 2000). Por tal motivo, este proceso se enfoca en la percepción del consumidor y la localización de las marcas dentro de segmentos específicos de mercado.

1.1.2.2. Marketing mix

La mezcla de mercadeo o marketing mix es una actividad clave para lograr los objetivos de la estrategia de marketing. Consiste en la combinación de cuatro elementos: producto, precio, comunicación y distribución, los cuales se utilizan para producir la respuesta que se quiere en el mercado meta.

1.1.2.2.1. Producto

Es un conjunto de características y atributos tangibles e intangibles que satisfacen las necesidades de los clientes. Las empresas promocionan productos de acuerdo a los gustos o requerimientos de los compradores. Internet da valor agregado a través de la información, lo que permite la creación de empresas con nuevos productos y servicios adaptados a diferentes grupos de clientes.

Las personas relacionan personalidades, estereotipos o características humanas a productos, los cuales presentan múltiples atributos físicos, psicológicos e incluso aspectos sociológicos asociados al consumidor. En la mente del cliente se forma una imagen del producto que la mayoría de las veces es subjetiva y que no siempre coincide con aspectos técnicos reales. El consumidor sintetiza esta imagen en función de sus atributos y coloca en su mente las marcas en relación a las de la competencia. Los principales atributos de producto son los siguientes (Muñiz 2005):

- **Núcleo:** Es el conjunto de propiedades físicas, químicas y técnicas del

producto.

- **Calidad:** Se divide en dos partes. La calidad técnica es la valoración del núcleo en función de cierto tipo de estándares. La calidad percibida es la valoración que un cliente hace sobre un producto, la cual es subjetiva la mayoría de las veces.
- **Precio:** Es el valor de comercialización de los productos y servicios. (Ver sección 1.1.2.2.2)
- **Envase:** Elemento de protección del producto que junto al diseño tiene un gran valor promocional y de imagen.
- **Diseño:** Forma y tamaño que permiten la identificación del producto o la empresa y configura la propia personalidad del mismo.
- **Marca:** Conjunto de nombres y expresiones gráficas que facilitan la identificación del producto y permiten su recuerdo asociado a uno u otro atributo.
- **Servicio:** Conjunto de valores añadidos a un producto que permiten marcar las diferencias respecto a competencia.
- **Imagen:** Se divide en dos partes: La imagen del producto es la opinión específica que se crea en la mente del consumidor. La imagen de la empresa es la opinión global arraigada en la memoria del mercado.

1.1.2.2.2. Precio

Es el valor monetario asignado a un bien o servicio. Internet tiene influencia directa en la política de precios debido a que da valor agregado a los productos, reduce costos de publicidad y aumenta el número de clientes. La red ofrece a los compradores la posibilidad de comparar precios y ofertas, lo que obliga a la competencia a regular los precios continuamente.

1.1.2.2.3. Comunicación

La comunicación a través de los medios digitales es interactiva, bidireccional y económica. Los negocios electrónicos utilizan técnicas de mercadeo, publicidad que aprovechan medios como: buscadores, portales, comunidades virtuales, páginas especializadas y publicaciones digitales. Por tanto, los contenidos son fundamentales para una comunicación efectiva (Ver sección 1.1.3).

1.1.2.2.4. Distribución

El Internet se ha convertido en una red de distribución estratégica que provee múltiples ventajas competitivas. La cantidad de intermediarios se reduce y se obtiene un alcance de ventas a nivel global. Las tecnologías de la información permiten la creación de canales alternativos y complementarios a los canales tradicionales mediante comercios electrónicos, los cuales ofrecen a los visitantes una mayor variedad de productos y servicios.

1.1.2.3. Entorno del e-marketing

En entorno del e-marketing se refiere al conjunto de factores externos que influyen el comportamiento de los consumidores meta y de la interacción de los mismos con el marketing mix. Estos factores son de diversa naturaleza por ejemplo: Internet, el marco legal de un país, aspectos socioculturales de un segmento de mercado, entre otros.

Para un análisis completo del entorno del e-marketing se lo estudia conforme a los siguientes aspectos:

1.1.2.3.1. Entorno tecnológico:

El medio de comunicación del e-marketing está constituido fundamentalmente por una gran infraestructura tecnológica. A través de Internet y la telefonía móvil se rompen las barreras geográficas y temporales del marketing tradicional. De tal forma, los clientes pueden realizar transacciones a nivel mundial.

1.1.2.3.2. Entorno ético y legal

Tal como en el mundo real, el desenvolvimiento del e-marketing debe regirse bajo parámetros éticos y legales. La privacidad del cliente es imprescindible y se deben seguir normas como la de no compartir los datos personales ni enviar correo masivo a menos que sea con consentimiento previo. El marco legal provee las pautas necesarias para ejercer muchas actividades sobre la red. En el contexto del Ecuador, por ejemplo, se dispone de la Ley de Propiedad Intelectual y la Ley de Comercio Electrónico.

1.1.2.3.3. Entorno competitivo

Los servicios de valor agregado constituyen una gran ventaja competitiva dentro

del mercado. Las estrategias de competitividad mantienen la posición de la empresa y sus productos en la red. Las organizaciones deben determinar las necesidades, deseos e intereses de los mercados meta para satisfacer de manera más eficiente y efectiva que sus competidores. La velocidad de acoplamiento a nuevos requerimientos es de suma importancia para desplazar a la competencia y fortalecer la presencia en Internet.

1.1.2.3.4. Entorno económico

Las condiciones económicas determinan la distribución de riqueza entre los consumidores. Los clientes disminuyen o aumentan su poder de adquisición de acuerdo a la situación de su región, reflejada en diferentes factores como por ejemplo: el precio de la inflación, el sueldo mínimo vital por persona, el valor de riesgo país, entre otros. Estos y otros factores influyen directamente en las estrategias de marketing ya que permiten la identificación de diferentes oportunidades inversión.

1.1.2.3.5. Entorno social y cultural

Toda estrategia de mercado que se desenvuelve a nivel mundial debe superar ciertas barreras referentes al ámbito social de las culturas y naciones. Para esto es necesario la adopción de diferentes políticas orientadas a la individualización de mercados globales, de tal forma que se incremente la velocidad de asimilación difusión de productos y servicios.

1.1.2.4. Servicios electrónicos

Un servicio se define como el conjunto de actividades y beneficios que satisfacen los requerimientos del cliente sin contar con los bienes de consumo físicos (Dalrymple 1990). Por otra parte, el avance de las tecnologías de comunicación digital constituye la antesala de nuevas generaciones de servicios.

Dentro del contexto de los negocios electrónicos, los e-services son servicios de valor añadido diseñados para obtener beneficios empresariales y crear mejores oportunidades de negocio. Por otro lado, para la informática, e-service es una aplicación de software autónoma que soporta cierto tipo de procesos de una organización con el uso de protocolos estandarizados sobre Internet.

En los últimos años se ha visto la proliferación sobre la red de las industrias de

telecomunicaciones, educación, finanzas, cuidado de la salud, ciencia, entre otras. Sin embargo, la tecnología debe superar cierto tipo de limitaciones relacionadas con los siguientes aspectos:

- **Interacción humana:** Las interfaces de usuario no son tan amigables como la interacción humana directa. Este es un elemento fundamental para la mayoría de aplicaciones, especialmente para los servicios de medicina electrónica en donde la comunicación entre médico y paciente es fundamental.
- **Diseño centrado al usuario:** La mayoría de procesos de diseño de software son más orientados hacia la tecnología que hacia el ser humano. Por ejemplo, debido a la falta de mecanismos estándar, el usuario debe recordar un sinnúmero de contraseñas a causa de la gran cantidad de servicios electrónicos suscritos.
- **Fácil integración e interoperatividad mejorada:** Hasta hace unos años, la integración y la interoperabilidad de los e-services era un gran problema por la diversidad de estándares. El aparecimiento de los servicios Web constituye un paso importante para el desarrollo de plataformas de servicios electrónicos mejor conectadas.
- **Capacidad de comunicación robusta:** Con la proliferación de los e-services se requiere una mejor capacidad de comunicación entre la empresa y el cliente. Los métodos de comunicación tradicional deben ser complementados por nuevas tecnologías. Dentro de los próximos años la telefonía y voz sobre IP van a jugar un papel fundamental.
- **Seguridad:** Los sistemas de servicios electrónicos requieren una excelente infraestructura para garantizar una seguridad robusta. Una débil protección de información sensible puede causar grandes pérdidas económicas.

1.1.3. COMUNICACIÓN PARA E-MARKETING

El e-marketing basa su comunicación a través de medios digitales. Esto ha ocasionado la creación de nuevos modelos y técnicas de intercambio de información que toman mayor ventaja del Internet. La Tabla 1.1 muestra la diferencia entre la comunicación del marketing tradicional y el e-marketing. Dentro del contexto de las tecnologías de información existen tres aspectos de gran importancia: comunicación integrada, gestión de relaciones con el cliente,

promoción y publicidad sobre internet.

Marketing tradicional	E-marketing
Modelo de comunicación uno a muchos	Modelo de comunicación uno a uno
Mercadeo de masas (Modelo push)	Mercadeo individualizado o personalización de masas (Modelo pull)
Comunicación direccional	Comunicación bidireccional
El cliente como objetivo	El cliente como compañero de negocio
Segmentación	Comunicación

Tabla 1.1 Diferencias de la comunicación entre el marketing tradicional y el e-marketing (Chaffey 2005).

1.1.3.1. Comunicación integrada

Las ventajas de la comunicación sobre la red son las siguientes: interactividad, inteligencia, individualización e integración.

1.1.3.1.1. Interactividad

El cliente usualmente inicia el contacto mientras busca información en Internet. El vendedor puede dar una atención individualizada a través de su sitio Web. Una compañía puede obtener y almacenar la reacción de un individuo ante los productos, promociones y servicios de la empresa y al mismo tiempo sus preferencias, gustos y necesidades, que en futuros encuentros pueden ser tomados en cuenta para presentar y recomendar la información correcta y personalizada al cliente.

1.1.3.1.2. Inteligencia

El Internet constituye un medio para la investigación de mercado. Permite recolectar las opiniones y percepciones de los clientes respecto de los productos o servicios promocionados. La información se recopila de a través de diversos métodos de recolección de datos del consumidor.

Por otra parte, la navegación del usuario se registra en archivos históricos de transacciones que son analizados por sistemas denominados *Log file Analyser*, los cuales permiten descubrir patrones y tendencias útiles que sirvan como base para tomar decisiones futuras.

1.1.3.1.3. *Individualización*

Esta es una característica importante en la comunicación integral y trata sobre el marketing a la medida del individuo. El proceso de personalización es necesario para lograr una gestión de relaciones con el cliente en línea.

La personalización se logra por medio de sitios Web que administran los procesos de compra y post-compra. Además se utiliza la personalización masiva de contenidos dirigidos a grupos de clientes con intereses comunes. (§ 1.1.3.3)

1.1.3.1.4. *Integración*

Internet es un medio de comunicación muy versátil debido a que ofrece múltiples posibilidades. Por ejemplo: WWW, e-mail, chat, e-radio o videoconferencia. Además, la red se complementa con otros canales de comunicación tales como los *call centers* con lo que se logra un servicio más completo en el ámbito de atención al cliente.

1.1.3.1.5. *Técnicas de e-marketing*

Las técnicas tradicionales de marketing han evolucionado con el avance de la tecnología digital. Por otra parte, la comunicación es más personalizada e interactiva; motivo por el cual se puede obtener información del cliente de manera inmediata. Dentro de las principales técnicas de e-marketing se encuentran las siguientes:

- Database marketing
- Marketing one-to-one.
- Métodos de recolección de información del consumidor.

1.1.3.1.5.1. *Database marketing:*

Esta técnica utiliza las bases de datos para obtener conocimiento acerca del mercado meta y el negocio. Esta información se utiliza con diferentes propósitos, como por ejemplo:

- Hacer segmentaciones de mercado.
- Obtener información estadística.
- Hacer investigación de mercado con el uso de técnicas de minado de datos.

1.1.3.1.5.2. Marketing one-to-one:

La información de productos y servicios llega muy fácilmente hacia los clientes que lo requieran con el uso de nuevas tecnologías como la telefonía móvil. Las técnicas de marketing one-to-one se basan fundamentalmente en conocer al cliente para orientar las políticas del negocio. Por medio de Internet es posible acumular gran cantidad de información a través las múltiples alternativas de comunicación que ofrece. Las interacciones del usuario son capturadas para analizar sus preferencias y comportamiento. De esta forma los sitios Web son capaces de ofrecer contenidos efectivos y personalizados.

1.1.3.1.5.3. Métodos de recolección de información del consumidor:

Existen múltiples maneras de adquirir datos del cliente a través de Internet. Esta información es de suma importancia para realizar un buen estudio del mercado meta. Dentro de los principales métodos de recolección de información utilizados por el e-marketing se encuentran:

- **Formularios electrónicos:** Permiten conocer qué tipo de clientes acceden al sitio Web. De con un análisis posterior se puede definir perfiles que sirven para identificar necesidades y requerimientos generales.
- **Comunidades virtuales:** Es un conjunto de personas con objetivos comunes que comparten información a través de la red. Por ejemplo: grupos de noticias, foros, chats, listas de distribución.
- **Las cookies:** Son archivos de texto que un navegador de Internet almacena en una computadora. Contienen información principalmente de identificación del usuario, frecuencia de visitas, o movimientos de una persona dentro de una página Web.
- **Clickstreaming:** Esta técnica consiste en crear un historial de la navegación del sitio Web hecha por un cliente para definir su perfil o determinar sus intereses.
- **Agentes inteligentes:** Son sistemas sofisticados de software capaces de observar el medio que los rodea, desarrollar un conjunto de acciones y aprender de los resultados de dichas acciones. Pueden ser utilizados tanto por el cliente como por el vendedor con el objetivo de delegar transacciones, como por ejemplo compra o venta sobre la red. (Ver sección 1.3)

1.1.3.2. Promoción y publicidad sobre Internet

Las características de la comunicación integrada ofrecen enormes ventajas a las empresas. Las estrategias para promocionar y difundir los productos y servicios de una organización son parte fundamental del marketing, a continuación los aspectos más relevantes:

- El cliente busca publicidad y tiene el control de la misma.
- La publicidad no tiene límites de espacio y tiempo.
- Es posible obtener datos estadísticos como por ejemplo el número de personas que ven un anuncio o el porcentaje de usuarios que al ver un anuncio, hacen clic y se dirigen a la página destino.
- La investigación de mercado en línea favorece notablemente a la personalización de la promoción y publicidad.
- Existen parámetros que brindan información y pueden medirse como: página por la que acceden, el recorrido y tiempo que navega el cliente en el sitio, el país, la dirección IP, entre otros.

El formato más común para publicitarse en Internet es el *banner*, el cual es generalmente de forma rectangular y contiene imágenes, animaciones y mensajes publicitarios. En la actualidad el video y el sonido incrementan el nivel de efectividad de la publicidad. Las empresas también promocionan sus productos y servicios a través de motores de búsqueda, consultas de información, suscripciones, y ventanas *pop-up*.

1.1.3.3. Gestión de relaciones con el cliente

Después de que los consumidores han hecho una compra pueden producirse efectos de descontento. Aquí es en donde intervienen los procesos de gestión de relaciones con el cliente o CRM (*Custom Relationship Management*). Uno de los principales objetivos del CRM es fortalecer la imagen de una organización frente a los clientes y de esta forma conservar la lealtad del consumidor. Si el cliente aprecia el buen servicio es muy difícil que vaya donde el competidor. La satisfacción del cliente no siempre equivale a lealtad. Para lograr este objetivo es necesario agregar servicios de valor agregado.

Existen dos definiciones de CRM que van en función de distintos puntos de vista:

- Para el marketing, CRM es una estrategia de negocios para fortalecer las

relaciones con los clientes con el uso de la tecnología.

- Desde el punto de vista de la informática, CRM es una aplicación de software basada en procesos para gestionar las relaciones con el cliente.

Para la implementación de una estrategia de gestión de relaciones todos los procesos de negocios relacionados con el cliente deben ser replanteados, por lo que es necesario un cambio importante dentro de la organización. Por tal motivo, implementar una estrategia de este estilo es costoso; sin embargo, los beneficios son innumerables.

El concepto de CRM converge hacia el concepto e-CRM (*Electronical Custom Relationship Management*) cuando los canales de comunicación se orientan exclusivamente a medios electrónicos. Por ejemplo, el CRM tradicional tiene como canal el teléfono o el fax; mientras que el e-CRM utiliza tecnología móvil e Internet como medios de comunicación fundamentales.

1.1.4. MODELO DEL COMPORTAMIENTO DEL CONSUMIDOR

Saber por qué las personas compran es un aspecto extremadamente complejo. El comportamiento del consumidor comprende el conjunto de acciones que intervienen en la búsqueda, obtención y consumo de productos, así como también los procesos de decisión que involucran dichas acciones.

Existen múltiples factores que influyen al momento de realizar una compra; sin embargo, una manera estructurada y sencilla de representar el comportamiento del consumidor es a través de un modelo de procesos. La personalidad del cliente y múltiples variables de entorno constituyen las entradas del proceso de compra, el cual tiene como resultado la elección o inclinación hacia un bien de consumo. Específicamente, el modelo del comportamiento del consumidor está conformado de tres partes (Figura 1.4):

- Factores de influencia
- Proceso de compra
- Decisión de consumidor

1.1.4.1. Factores de influencia

Existen una gran variedad de variables que intervienen en el proceso de compra. Los tipos de factores de influencia son de dos clases: los estímulos externos que

proviene del entorno del consumidor y los estímulos internos que están conformados por las características psicológicas del mismo.

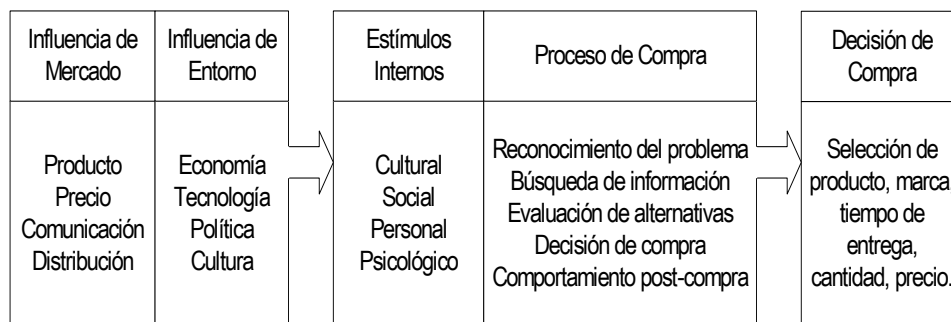


Figura 1.4 Modelo de comportamiento del consumidor (Pressley 2003).

1.1.4.1.1. *Estímulos externos*

El consumidor está expuesto a diferentes influencias que pertenecen al ambiente en el que se desenvuelve. Existen dos clases de estímulos externos: los estímulos de mercado y los estímulos de entorno. Los estímulos de mercado son todos aquellos que están relacionados con el marketing mix, por ejemplo el tipo de producto o servicio, el precio, la ubicación del almacén y la publicidad. Por otro lado, los estímulos de entorno comprenden factores que son de tipo económico, tecnológico, político y socio-cultural, por ejemplo: status económico, nacionalidad, nivel de educación, entre otros.

1.1.4.1.2. *Estímulos internos*

La personalidad es el conjunto de cualidades que caracterizan al individuo. Los estímulos internos que influyen el proceso de compra son elementos de la personalidad del consumidor como por ejemplo: experiencias pasadas, información almacenada en la memoria, gustos o aficiones.

1.1.4.2. **Proceso de compra**

Este proceso describe de manera formal todas las actividades involucradas en el transcurso de la adquisición de bienes de consumo. El consumidor reconoce un problema al identificar una necesidad o requerimiento. Luego, procede a buscar información con el objetivo de encontrar alternativas de solución; esto se lo hace al recordar experiencias pasadas o al conversar con otras personas. Aquí la publicidad juega un papel fundamental.

Una vez que se ha recolectado información suficiente, se hace una evaluación y

selección de las alternativas más convenientes antes de hacer la transacción de compra en sí. Finalmente, el cliente acumula experiencia con el uso de un producto o servicio, lo que es de gran utilidad ya que esta información sirve en futuras ocasiones.

El proceso de compra está conformado por las siguientes etapas:

1. Reconocimiento del problema o necesidad.
2. Búsqueda de alternativas.
3. Evaluación de alternativas.
4. Compra.
5. Post-compra.

En la Figura 1.5 se observa un esquema simple del proceso de compra en donde estas etapas son secuenciales y van en función del tiempo. Cabe mencionar que algunas veces se pueden solapar e incluso se pueden generar ciclos.

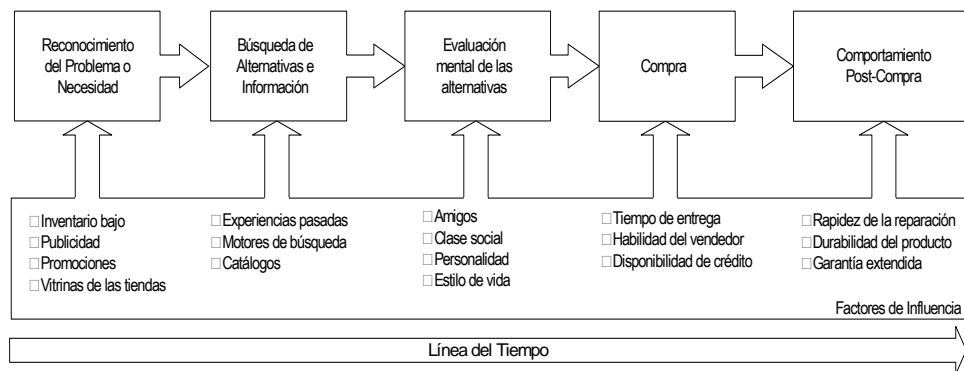


Figura 1.5 Proceso de compra del consumidor (Dalrymple 1990).

1.1.4.2.1. Reconocimiento del problema o necesidad

En esta etapa el consumidor se da cuenta de la existencia de un requerimiento, problema o necesidad. El reconocimiento del problema puede ser activado tanto por factores internos como experiencias almacenadas en la memoria y cierto tipo de motivaciones básicas.

También existen factores externos como la publicidad y los grupos de referencia a los que pertenece el consumidor. La publicidad es un elemento que tiene gran importancia y se lo describe en la sección 1.1.3.2.

Es necesario mencionar que algunas de las necesidades de la persona se manifiestan inconscientemente y a veces puede pasar cierta cantidad de tiempo hasta que se reconozca de manera consciente que verdaderamente existe un

problema y se proceda a la búsqueda de una solución.

1.1.4.2.2. Búsqueda de alternativas

El consumidor queda involucrado en un proceso de resolución de problemas después haber identificado sus necesidades. El primer sitio en donde la se busca información para resolver el problema es en la memoria. El consumidor se vale de experiencias pasadas y recuerdos adquiridos a través de publicidad, comentarios, recomendaciones, entre otros. Si la información en la memoria no da los medios para solucionar el problema entonces el consumidor acude a fuentes de información externas como por ejemplo vendedores, referencias de otras personas, catálogos de productos, Internet. Toda esta información es sintetizada en la memoria y pasa por una fase de filtrado en donde es distorsionada o descartada. Este proceso de interpretación de los datos se le llama percepción.

1.1.4.2.3. Evaluación de alternativas

Ésta es una de las fases más complejas de comprender debido a que es difícil saber lo que sucede dentro de la mente del consumidor. Al evaluar diferentes alternativas se compara la información a través de un proceso llamado criterio evaluativo, el cual tiene como resultado un conjunto de creencias u opiniones acerca de las consecuencias de diferentes alternativas de compra posibles. Estas opiniones se almacenan en la memoria y se combinan para sustentar cierto tipo de actitudes que predisponen al consumidor a la compra de determinado bien. Las actitudes positivas están fuertemente relacionadas con la intención de compra de producto. Una actitud negativa puede retardar la compra o no efectuarla. Medir la intención de compra resulta sumamente importante ya que involucra la probabilidad de que una compra sea hecha.

1.1.4.2.4. Compra

El consumidor realiza la compra después de haber elegido alguna alternativa interesante. Sin embargo es necesario aclarar que en este proceso influyen varios factores como por ejemplo la intención de compra, la ubicación de la tienda, la disponibilidad de crédito, incluso la habilidad del vendedor.

Por ejemplo, un comprador puede tener la intención de comprar una marca determinada; sin embargo, si se acabó el stock del producto es posible que elija

otra marca o no hay el suficiente dinero para comprar la marca deseada.

Las decisiones de compra pueden ser influidas por un sin número de elementos y factores que dependen del tipo de compra. No es lo mismo hacer una compra directamente en un almacén que hacerla a través de Internet. Los precios especiales constituyen un fuerte estímulo para motivar una compra o para cambiar de marca. La diferencia de precio es fundamental cuando el cliente percibe que las diferentes marcas son casi de la misma calidad.

Un personal altamente capacitado en ventas puede ser de gran ayuda para que el cliente se incline por cierto tipo de bien de consumo. Los vendedores que conocen el producto y pueden explicar sus características pueden negociar mejor y llegar a un trato con el cliente. Por tanto, todas las compañías que capacitan a su personal de ventas tienen mayor probabilidad compra por parte del consumidor.

1.1.4.2.5. Comportamiento post-compra

El proceso de compra finaliza con la fase de post-compra. Esta etapa provee de información que retroalimenta el proceso. El cliente aprende más acerca de los productos o servicios al comparar las expectativas iniciales con la experiencia real; lo que puede causar contento o descontento en la actitud del cliente quien tiene diferentes reacciones. Si un cliente está satisfecho, su opinión acerca de un producto será favorable, lo que aumenta la probabilidad que lo compre de nuevo o lo recomiende a otra persona.

1.1.4.3. Decisión del consumidor

El principal producto del proceso de compra del consumidor es una decisión que está directamente relacionada con detalles de la compra propiamente dicha. Estas decisiones pueden variar desde la selección de un tipo de marca, producto o servicio hasta detalles referentes al momento y cantidad de compra a efectuar. Dentro de los principales tipos de decisión que un cliente puede realizar se encuentran: selección de producto, marca o proveedor, momento de compra y cantidad.

El objetivo de un gerente de ventas es tratar de orientar esta decisión mediante las herramientas que provee el marketing.

1.2. ONTOLOGÍAS

En el contexto de la filosofía se define ontología como la descripción de lo que existe. Históricamente, este término ha tenido diferentes significados; sin embargo, los enfoques filosóficos contemporáneos utilizan herramientas matemáticas como la lógica formal para analizar las categorías fundamentales de objetos y estudios de las relaciones de dependencia entre la parte y el todo.

El campo de las ciencias de la computación ha adoptado este tópico para estudiarlo desde una nueva perspectiva que contempla cuestiones relacionadas con representación del conocimiento. Además, es necesario señalar que las ontologías se han convertido, en los últimos años, en el foco de investigación de diversas comunidades científicas debido a su gran potencial de aplicación a través de las tecnologías de la información. Los principales beneficios se basan en la simplificación de tareas relacionadas con la representación, intercambio y reutilización del conocimiento. De esta manera se da un nuevo matiz a todas las posibilidades de comunicación entre los sistemas de información y sus usuarios al mejorar su capacidad de interacción.

1.2.1. ASPECTOS GENERALES

Dentro del ámbito de la inteligencia artificial algunos términos pueden causar confusión debido a que su contexto está íntimamente relacionado. Por esta razón es necesario comprender las siguientes definiciones:

- **Metadatos:** Definidos como datos acerca de los datos, constituyen la información acerca de un conjunto de objetos. Tienen amplia variedad de aplicaciones en bases de datos, *data warehouse* y búsquedas sobre la Web.
- **Red semántica:** Técnica de representación del conocimiento que consiste en un grafo dirigido cuyos nodos representan conceptos y sus arcos representan relaciones semánticas entre dichos conceptos.
- **Taxonomía:** Es una estructura que organiza términos de un vocabulario controlado de manera jerárquica. Formalmente se define como un árbol de clasificación de un conjunto de objetos; en el nodo raíz se encuentra la categoría máxima mientras que los nodos de los niveles inferiores representan categorías específicas.

- **Dominio del conocimiento:** Es un conjunto de información estrechamente relacionada que está presente en un contexto del mundo real y despierta el interés de una comunidad determinada (Gonçalves 2004). En un dominio del conocimiento se encuentran taxonomías cuyos componentes generan soluciones a un conjunto de problemas. Cuando está representado formalmente se lo denomina dominio de discurso, universo de discurso o dominio de interpretación.
- **Web Semántica:** Constituye la generación sucesora de la Web actual en donde la información se presenta junto con metadatos de contenidos semánticos procesables por servicios de razonamiento. De esta forma se incrementan los niveles de reutilización de datos, automatización de procesos e integración de los sistemas informáticos para potenciar ampliamente la utilidad del Internet y la Web.

1.2.1.1. Definición

Ontología es una especificación explícita de una conceptuación¹ (Gruber 1993a). Entiéndase conceptuación como una abstracción simplificada del mundo real. Todo sistema de información está sometido a cierto nivel de conceptuación que puede ser implícito o explícito como en el caso de los sistemas procedimentales y los sistemas basados en conocimiento respectivamente.

La manera de especificar una conceptuación es mediante formalismos matemáticos; por tal motivo la definición anterior queda extendida al establecer que una ontología informática es una especificación de un vocabulario formal que representa el significado de conceptos que pertenecen a un dominio de discurso al definir clases, relaciones, funciones y otros elementos involucrados. En la Figura 1.6 se muestra un ejemplo gráfico.

Finalmente, es importante resaltar que una ontología es una técnica de representación del conocimiento que puede ser expresada mediante una lógica de primer orden y por tanto puede ser trasladada a un lenguaje simbólico procesable por máquina.

¹ Traducido del inglés *conceptualization*. La palabra conceptualización no existe en castellano.

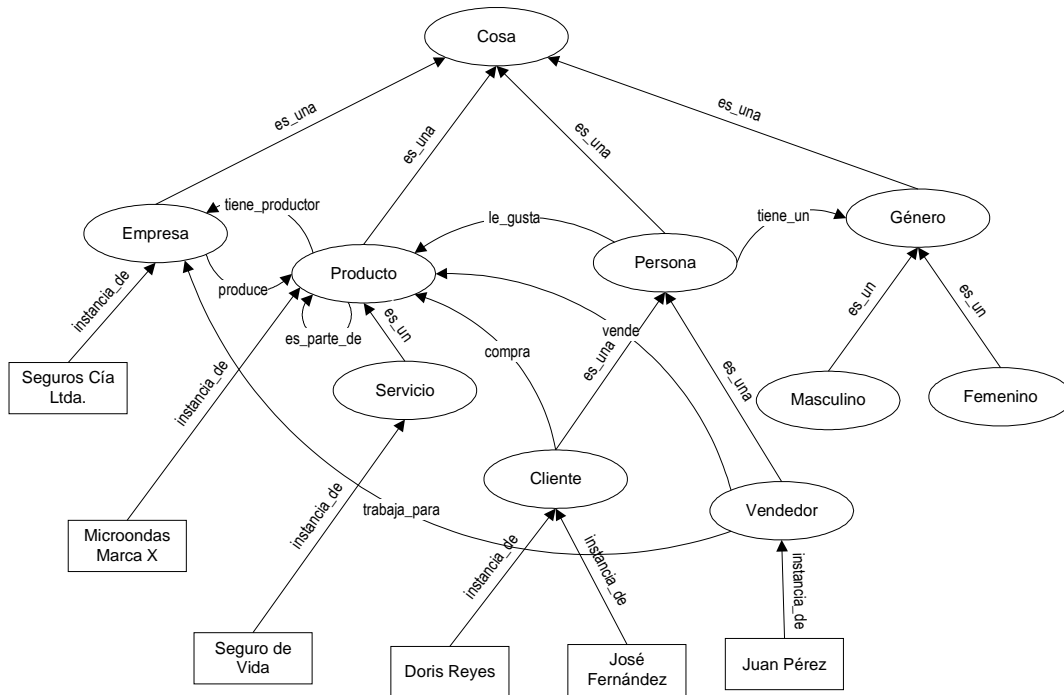


Figura 1.6 Esquema simplificado de una ontología de un dominio de ventas. (Los Autores).

1.2.1.2. Tipos de ontologías

De acuerdo al contenido se clasifican en: ontologías generales, de dominio, de tarea, de aplicación y de representación. La Figura 1.7 muestra el nivel de dependencia de cada uno de estos tipos. Las ontologías de representación quedan embebidas dentro de las ontologías generales. En la Tabla 1.2 se listan algunos ejemplos de ontologías desarrolladas.

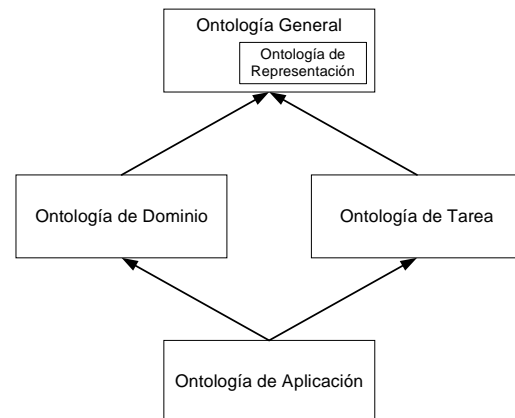


Figura 1.7 Tipos de ontologías y su nivel de dependencia (Los Autores).

- **Ontologías generales:** Son conceptualizaciones que son independientes de un problema o dominio en particular. Describen conceptos generales relacionados con espacio, tiempo, materia, objeto, evento, acción entre otros.
- **Ontologías de dominio:** Expresan conceptualizaciones de dominios particulares por ejemplo medicina o derecho.
- **Ontologías de tarea:** Expresan conceptos relacionadas con resolución de problemas independientes del dominio en que se desempeñan como por

ejemplo tareas de diagnóstico, diseño y clasificación.

- **Ontologías de aplicación:** Describen conceptos en función de una ontología de dominio y tarea particulares. Estos conceptos corresponden a entidades del dominio que se encuentran relacionadas con una actividad o aplicación específica.
- **Ontologías de representación:** También conocidas como meta-ontologías, describen conceptos relacionados a formalismos de representación del conocimiento.

Tipo de ontología	Ontología	Descripción	Sitio Web
General	OpenCYC	Base de conocimiento general	www.opencyc.org
	SUO	Ontología General Estándar (Standard Upper Ontology)	suo.ieee.org
Dominio	UMLS	Sistema de Lenguaje Médico Unificado (Unified Medical Language System)	www.nlm.nih.gov/research/umls/umlsmain.html
	RossetaNet	Partes de computadores	www.rosettanet.org
	UNSPSC	Clasificación Universal Estándar de Productos y Servicios (Universal Standard Product and Services Classification)	www.unspsc.org
Tarea	PSM	Métodos de Resolución de Problemas (Problem Solving Methods)	protege.stanford.edu/plugins/psmtab/PSMTab.html

Tabla 1.2 Ejemplos de ontologías (Los Autores)

1.2.1.3. Elementos de una ontología

Las ontologías constituyen una técnica de representación del conocimiento que sintetiza y extiende otras técnicas. Ésa es la razón por la cual existen gran cantidad de elementos en común con objetos, redes semánticas, plantillas, entre otros. Los elementos de una ontología son: clases, relaciones, propiedades, restricciones, funciones, axiomas e instancias.

1.2.1.3.1. Clases

Son representaciones de objetos reales o abstractos que pertenecen a un dominio del conocimiento. Dentro de una taxonomía de clases, a los conceptos de nivel inmediato superior se los denomina superclases mientras que a los conceptos de

nivel inmediato inferior se los denomina subclases.

1.2.1.3.2. Relaciones

Son la interconexión de dos clases que representa la interacción entre dos objetos. Estas relaciones son de carácter semántico. En la Tabla 1.3 se describen los principales tipos de relaciones semánticas.

Tipo de Relación	Significado	Etiqueta
Meronimia	A es parte de B	parte-de (part-of)
Holonimia	B tiene A como parte de sí mismo	tiene-un (has-a), tiene-parte (has-part), todo-de (whole-of)
Hiponimia	A es subordinado de B	es-un (is-a), tipo-de (kind-of)
Hipernimia	A es superordinado de B	—
Sinonimia	A denota lo mismo que B	sinónimo-de (synonym-of)
Antonimia	A es opuesto a B	antónimo-de (antonym-of)

Tabla 1.3 Principales tipos de relaciones semánticas (Wikipedia 2005).

1.2.1.3.3. Propiedades

Son los atributos y características que describen cada concepto. Se les conoce también como roles o *slots*.

1.2.1.3.4. Restricciones

Son el conjunto de características y limitaciones del valor de las propiedades. Dentro de los principales tipos de restricciones se encuentran: tipo de valor, valores permitidos y cardinalidad.

- Los tipos de valores que una propiedad puede tener son valores numéricos, valores lógicos, cadenas de texto, clases e instancias de otras clases.
- Los valores permitidos dependen del tipo de valor y se pueden expresar a través de conjuntos de clases, listas de etiquetas o rangos en caso de valores numéricos.
- La cardinalidad es el número de valores que puede contener una propiedad.

1.2.1.3.5. Funciones

Son un tipo concreto de relaciones donde se identifica un elemento mediante un

cálculo que considera varios elementos de la ontología (Lozano 2001).

1.2.1.3.6. Axiomas

Son proposiciones lógicas que se declaran sobre los elementos de la ontología, a partir de las cuales se infieren otras proposiciones o conocimiento que está implícito. Por ejemplo: "Si A es de clase B y B es de clase C entonces A es de clase C".

1.2.1.3.7. Instancias

Representan los conceptos más específicos pertenecientes a las clases. Una ontología y su respectivo conjunto de instancias constituyen una base de conocimiento.

1.2.2. CRITERIOS DE DISEÑO Y PROCESO DE CONSTRUCCIÓN DE ONTOLOGÍAS

1.2.2.1. Criterios de diseño

El diseño de una ontología requiere combinar diferentes criterios en donde se sacrifican unos con otros en función de diversos requerimientos o necesidades para lograr una representación del conocimiento efectiva. Los criterios fundamentales para diseño de ontologías son (Gruber 1993b):

1.2.2.1.1. Claridad

Es la relación que existe entre el vocabulario empleado y los conceptos a ser representados. Una ontología es clara y objetiva cuando el significado de los conceptos es descrito apropiadamente por los términos del vocabulario, lo que permite un entendimiento claro por parte de los usuarios de la ontología.

1.2.2.1.2. Coherencia

Los axiomas y definiciones deben ser lógicamente consistentes para garantizar que las inferencias realizadas sobre la ontología sean correctas.

1.2.2.1.3. Extensibilidad

Maximiza la capacidad de extender una ontología a través de la definición de nuevos términos especializados sin la necesidad de una revisión o modificación

de las definiciones existentes.

1.2.2.1.4. Independencia del lenguaje de codificación

La concepción se debe especificar a un nivel en el que no dependa en mayor grado de un lenguaje de representación en particular, de tal forma que sistemas basados en diferentes lenguajes sean capaces de compartir información.

1.2.2.1.5. Nivel de dependencia

Una ontología debe tener el suficiente nivel de generalización o especialización en un ambiente de intercambio de información, de tal forma que sus usuarios tengan la suficiente libertad de extender o crear instancias conforme lo necesiten y de esta forma mantener un nivel de comunicación consistente.

1.2.2.2. Proceso de construcción de ontologías

Al considerar los criterios de diseño, pueden surgir diversos procesos de construcción de ontologías. Cabe anotar que no hay un procedimiento convencional para el desarrollo de ontologías por lo que se presenta a continuación un esquema simple conformado por los siguientes pasos:

1. Determinar el dominio y alcance de la ontología.
2. Considerar el uso de ontologías existentes.
3. Enumerar términos importantes.
4. Definir la jerarquía de clases.
5. Definir las propiedades de las clases.
6. Definir las restricciones de las propiedades.
7. Crear instancias.

1.2.2.2.1. Determinar el dominio y alcance de la ontología

Para determinar precisamente el dominio y alcance es importante responder a las siguientes preguntas básicas acerca de la ontología a desarrollar:

- ¿Cuál es el dominio que cubrirá?
- ¿Para qué será utilizada?
- ¿Qué tipos de preguntas deberá responder?
- ¿Quién la utilizará y dará mantenimiento?

Las respuestas de estas preguntas pueden variar durante el proceso de diseño,

sin embargo son de gran utilidad para limitar el alcance del modelo. Otra forma para definir el alcance y el dominio es a través de un banco de preguntas específicas cuyas respuestas debe responder la ontología.

1.2.2.2. Considerar el uso de ontologías existentes

La reutilización de ontologías previamente desarrolladas ahorra gran cantidad de tiempo. En la mayoría de casos es necesario considerar extender y refinar recursos existentes para la representación de una tarea o dominio en particular. Existen múltiples librerías disponibles en la Web. La mayoría de ambientes de desarrollo de ontologías son capaces de importar y exportar ontologías de diferentes formatos.

1.2.2.2.3. Enumerar términos importantes

Es importante listar todos los términos involucrados en el dominio del conocimiento a ser representado. En primera instancia no es necesario diferenciar si representan clases, propiedades o relaciones. De esta manera se puede obtener un vocabulario base para hacer una selección de los términos más apropiados.

1.2.2.2.4. Definir la jerarquía de clases

Los métodos para desarrollar una jerarquía de clases son:

- El proceso de desarrollo *top-down* empieza con la definición de los conceptos más generales en el dominio hacia los conceptos más específicos.
- El proceso de desarrollo *bottom-up* empieza con la definición de los conceptos más específicos a los más generales.
- El proceso que combina los dos anteriores.

La selección de uno de estos métodos depende del nivel de conocimiento acerca del dominio. Los términos se ordenan a manera de clases y subclases para formar una estructura jerárquica. Cabe anotar que las estructuras taxonómicas son una parte fundamental de la ontología.

1.2.2.2.5. Definir las propiedades de las clases

Las clases por si solas no proveen suficiente información para responder las

preguntas planteadas inicialmente, se debe definir la estructura interna de los conceptos a través de sus propiedades.

Las propiedades pueden ser intrínsecas o propias de la clase (ej.: color, tamaño o sabor) y extrínsecas que cumplen un rol dentro de la ontología (ej.: nombre, código); incluso representar relaciones entre clases. Todas las subclases de una clase heredan sus propiedades, pero adicionalmente en la subclase se definen propiedades particulares que la diferencian de otras subclases.

1.2.2.2.6. *Definir las restricciones de las propiedades*

Consiste en definir las reglas que restringen en valor de las propiedades. De esta manera se garantiza la consistencia de la información que adquieren las instancias.

1.2.2.2.7. *Crear instancias*

Es el último paso en la creación de ontologías. Al añadir instancias individuales a la jerarquía de clases se obtiene una base de conocimiento funcional.

1.2.3. LENGUAJES DE REPRESENTACIÓN

Existen diferentes lenguajes para representar ontologías los cuales varían en notación; sin embargo, los estándares basados en XML han tomado gran importancia debido a la aparición de la Web semántica. En la Figura 1.8 se muestra el nivel de dependencia de estos lenguajes.

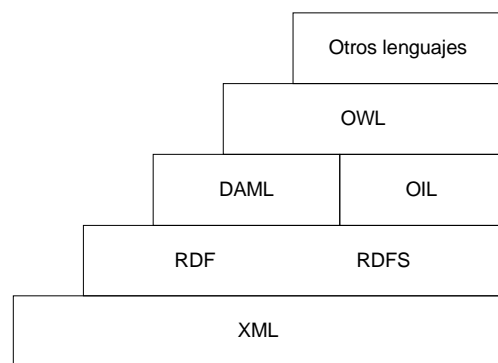


Figura 1.8 Nivel de dependencia de lenguajes de representación basados en XML.

1.2.3.1. XML

El lenguaje de marcado extensible (eXtensible Markup Language) permite una escritura de documentos estructurada con un vocabulario definido por el usuario. Existen dos métodos para restringir la sintaxis de un XML:

- Definición del tipo del documento o DTD (Document Type Definition) define la estructura, jerarquía y el orden de los elementos de un XML.
- Esquema XML tiene la funcionalidad de un DTD y además define tipos de datos y reglas de sintaxis complejas.

Un documento XML está diseñado para representar metadatos por medio de etiquetas personalizadas para facilitar la comunicación estandarizada entre sistemas de información.

1.2.3.2. RDF

El lenguaje de descripción de recursos (Resource Description Framework) es un estándar definido sobre XML que tiene como propósito representar los objetos de la Web. Permite que las aplicaciones intercambien información y soporta procesos con metadatos. El RDF está conformado por tres elementos fundamentales: recursos, propiedades y declaraciones.

- Los recursos describen objetos individuales de algún tipo, por ejemplo documentos electrónicos, servidores, usuarios, entre otros. Cada recurso tiene un identificador universal URI (Universal Resource Identifier), por convención se utilizan generalmente direcciones URL.
- Las propiedades expresan aspectos específicos, características, atributos o relaciones entre recursos.
- Las declaraciones están conformadas por un recurso específico, una propiedad y su valor asociado.

Por otro lado, el esquema RDF (RDFS) tiene la capacidad de describir vocabularios a través de la declaración de clases de recursos y sus propiedades asociadas. De esta forma se obtiene una representación estándar de redes semánticas que posteriormente pueden ser utilizadas como bases de conocimiento por sistemas inteligentes.

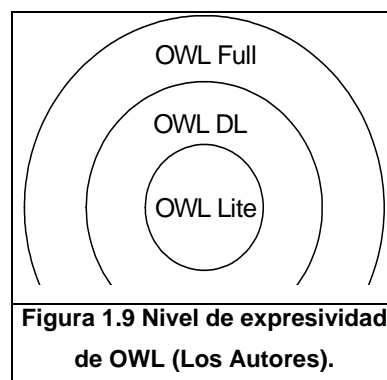
1.2.3.3. OWL

Es un lenguaje para definir y crear instancias de ontologías sobre la Web (Ontology Web Language). Proviene de la combinación de DAML (DARPA Agent Markup Language) y OIL (Ontology Inference Layer) que son lenguajes orientados a semántica definidos sobre RDF. Como resultado se obtuvo un lenguaje expresivo con una sofisticada clasificación de las estructuras de datos y propiedades.

OWL está basado en lógica de descripción la cual es un tipo de lógica de primer orden orientada a la representación semántica.

Existen tres tipos de OWL con diferente nivel de expresividad (Figura 1.9):

- OWL Lite soporta requerimientos básicos de clasificación jerárquica y especificación de restricciones.
- OWL DL soporta todos los elementos de la lógica de descripción para garantizar la efectividad de sistemas de razonamiento.
- OWL Full permite la libertad sintáctica de RDF sin embargo sacrifica la eficiencia y efectividad de los sistemas de razonamiento.



OWL es el lenguaje por excelencia para la representación del conocimiento para la Web Semántica. Además, puede ser utilizado como estándar de comunicación entre agentes inteligentes y sistemas distribuidos.

1.2.4. APLICACIONES EN EL ÁMBITO DEL E-BUSINESS

El gran número de aplicaciones de las ontologías está relacionado en mayor o menor grado con la comunicación. Dentro del ámbito de los negocios electrónicos existen muchos aspectos que pueden ser mejorados gracias a esta tecnología. En la Tabla 1.4 se presentan algunos usos para comercio electrónico.

Aspecto	Aplicación
Productos	Representación de atributos de productos complejos. Estandarización de catálogos de productos. Representación de funcionalidades de productos.
Clientes	Representación de perfiles complejos de usuario. Representación de intereses y gustos del cliente.
Negocio	Representación de esquemas de negocio. Representación de estrategias de e-marketing

Tabla 1.4 Aplicaciones de ontologías para comercio electrónico (Los Autores)

Las principales aplicaciones de las ontologías para e-business son:

- Formalización de procesos de negocio.
- Incremento de la interoperabilidad organizacional.
- Formalización de la comunicación en el negocio.
- Formalización de la base del conocimiento corporativa.
- Minado de datos y descubrimiento del conocimiento.

1.2.4.1. Formalización de procesos de negocio

Un aspecto fundamental para el éxito de una organización radica en la disciplina y formalidad con la que se aplican sus procedimientos. A través de la de la definición de una ontología de los proceso del negocio se incrementan los niveles de automatización, lo que aporta a que la organización se concentre en sus objetivos.

1.2.4.2. Incremento de la interoperabilidad organizacional

Entre unidades organizacionales existe un cierto nivel de comunicación a pesar de tener vocabularios diferentes de acuerdo a su cultura, experiencia, y objetivos. Al definir un sistema formal de comunicación, se asegura que las diferencias de terminología no afecten en mayor grado el intercambio de información, con lo cual se mejora la eficiencia de la organización.

1.2.4.3. Formalización de la comunicación en el negocio

El éxito del funcionamiento de las unidades de una organización puede ser limitada por la falta de claridad en la comunicación entre los diferentes niveles de dirección. Una ontología con la estructura genérica del negocio establece un contexto tanto para los procesos automatizados como para las personas, con lo que se minimiza la ambigüedad de la comunicación.

1.2.4.4. Formalización de la base del conocimiento corporativa

El conocimiento acumulado a lo largo del tiempo es esencial para el mejoramiento de procesos y decisiones futuras de organización. Con la estructuración semántica de registros, bitácoras y experiencias se obtiene una memoria corporativa que es de gran utilidad para facilitar la capacitación del personal y evitar errores del pasado.

1.2.4.5. Minado de datos y descubrimiento del conocimiento

Una conceptualización que incluya estructura de las base de datos de una organización ahorra gran cantidad de tiempo en procesos de descubrimiento de conocimiento. En contraste con otras técnicas, las relaciones semánticas de la ontología ponen en evidencia patrones importantes en las fases iniciales del proceso de minado de datos lo que permite la obtención de información más útil.

1.3. AGENTES INTELIGENTES

La tecnología de agentes inteligentes es producto de la síntesis de todas las disciplinas de la inteligencia artificial. La aplicación de esta tecnología en el campo de los negocios electrónicos automatiza un gran número de tareas que necesitan de la intervención humana. Por esta razón, los agentes que soportan los procesos de e-commerce necesitan incluir el conocimiento de expertos como compradores, vendedores y especialistas de marketing. De esta manera se pueden conseguir de una forma más efectiva los objetivos de un negocio electrónico. Consecuentemente, las actividades comerciales sobre Internet cuentan con mejores servicios personalizados y eficientes.

1.3.1. ASPECTOS GENERALES

Dentro del ámbito de los agentes inteligentes es necesario tomar en cuenta las siguientes definiciones:

- **Sensor:** Mecanismo o artificio utilizado para determinar el valor actual de una magnitud controlada de un sistema.
- **Efector:** Mecanismo o artificio utilizado para modificar el valor actual de una magnitud controlada de un sistema.
- **Percepción:** Información resultante de las impresiones obtenidas mediante sensores.
- **Acción:** Impresión o influencia que produce la actividad de una cosa en otra.
- **Agente:** Entidad que percibe su ambiente mediante sensores y que responde o actúa en tal ambiente mediante efectores; por ejemplo: agentes biológicos y agentes artificiales (Russell 1996).
- **Agente de software:** Es un agente artificial que opera en un ambiente de software.

1.3.1.1. Definición

Agente inteligente es un agente de software que ejecuta una tarea específica utilizando información recolectada del ambiente, para actuar de manera apropiada hasta completar dicha tarea de manera exitosa; además, debe ser capaz de autoajustarse al percibir cambios del ambiente de forma que produzca el resultado

esperado (Jiménez 2000). En la Figura 1.10 se presenta un esquema de un agente y la interacción con su entorno.

Se considera que los agente inteligentes son racionales en base a cuatro factores que permiten evaluar su desempeño y son los siguientes (Russell 1996):

- La medida con la que se evalúa el éxito logrado.
- El historial de percepciones del agente (secuencia de percepciones).
- El conocimiento que el agente posee del entorno en donde opera.
- Las acciones que el agente puede emprender.

Consecuentemente, un agente inteligente racional ideal es aquel que ejecuta las acciones que permiten obtener la máxima medida de rendimiento en todos los casos posibles de secuencias de percepciones. Para el logro de tal objetivo, el agente debe basarse en la información aportada por la secuencia de percepciones y el conocimiento incorporado en el agente.

1.3.1.2. Características

Un agente inteligente se diferencia de un sistema de software debido a que posee en mayor o menor grado ciertas características que dependen del tipo de agente y sus funciones. Estas características se detallan a continuación:

- **Autonomía:** Capacidad de interactuar con el entorno y tomar decisiones sin la intervención de los humanos u otros agentes.
- **Sensibilidad:** Capacidad para comunicarse con otros agentes o aplicaciones a través de un lenguaje en particular.
- **Reactividad:** Este atributo permite que el agente actué en reacción a los cambios que percibe del entorno.
- **Pro-actividad:** Es la capacidad de comportarse de acuerdo a los objetivos planteados.
- **Continuidad:** Los agentes están constantemente realizando procesos tanto al percibir información del entorno como al ejecutar acciones.

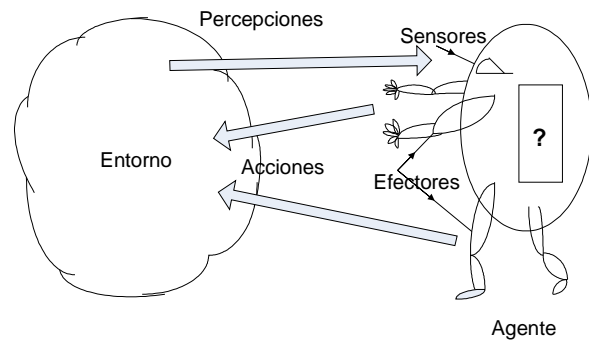


Figura 1.10 Esquema de un agente inteligente (Julián 2000).

- **Benevolencia:** Es la capacidad de satisfacer solicitudes de información por sí mismo o proveer información a otros agentes sin entrar en conflicto con sus objetivos.
- **Racionalidad:** El agente realiza las acciones necesarias para satisfacer sus objetivos.
- **Adaptabilidad:** Es la capacidad de aprender de su entorno; por ejemplo: preferencias de usuario, fuentes de información y otros agentes.
- **Movilidad:** El agente tiene la capacidad de trasladarse de forma auto-dirigida de una plataforma a otra.

1.3.1.3. Estructura de un agente

La estructura de un agente corresponde a la siguiente fórmula:

$$\text{Agente} = \text{Programa del agente} + \text{Arquitectura}$$

La arquitectura corresponde al sistema de computación en el que se ejecuta el agente; puede variar desde una computadora hasta algún tipo de hardware específico como por ejemplo cámaras de video para procesamiento de percepciones.

El programa de agente está compuesto por varios elementos como los sensores y efectores que procesan las percepciones y acciones respectivamente. Además, intervienen otros elementos como las metas a alcanzar, las medidas de desempeño y el ambiente en donde va a interactuar el agente.

A estos elementos se los denomina PAMA cuyas siglas significan Percepciones, Acciones, Metas y Ambiente. En la Tabla 1.5 se presentan varios ejemplos de agentes con su PAMA.

Tipo de agente	Percepciones	Acciones	Metas	Ambiente
Sistema de diagnóstico	Síntomas, evidencias, respuestas del paciente	Preguntas, exámenes, tratamientos	Salud del paciente, Reducción de costos	Paciente, hospital
Sistema de análisis de imágenes satelitales	Píxeles de intensidad variable, color	Clasificación de escena	Clasificación correcta	Imágenes satelitales

Controlador de refinería	Lecturas de temperatura y presión	Abrir y cerrar válvulas, ajustar temperatura	Maximizar la pureza, producción, seguridad	Refinería
Tutor interactivo de inglés	Palabras escritas	Imprimir ejercicios, sugerencias, correcciones	Maximizar el puntaje de los estudiantes en las pruebas	Conjunto de estudiantes

Tabla 1.5 Ejemplos de agentes y la descripción de su PAMA. (Russell 1996)

1.3.1.4. Tipos de ambiente

El ambiente en donde un agente interactúa tiene diferentes características que influyen directamente en el diseño y funcionamiento del mismo. Los tipos de ambiente (Russell 1996) se listan a continuación y en la Tabla 1.6 se presentan algunos ejemplos:

- **Accesible o inaccesible:** Un ambiente del cual se puede percibir información de su estado de manera completa, rápida y precisa es accesible, caso contrario es inaccesible.
- **Determinista o no determinista:** Si el estado siguiente del entorno es completamente determinado en función del estado actual y la acción escogida por el agente, el entorno es determinista, caso contrario es no determinista.
- **Episódicos o no episódicos:** La experiencia del agente se la puede dividir en episodios que representan relaciones entre percepciones y acciones. La calidad de las acciones en un ambiente episódico no depende de la acción realizada anteriormente por el agente, por tal razón solo se considera el estado actual del entorno para ejecutar una acción.
- **Estáticos o dinámicos:** Si el ambiente cambia mientras el agente opera es dinámico, por el contrario si no cambia es estático.
- **Discretos o continuos:** Si existe un número finito de acciones y percepciones el entorno es discreto, caso contrario es continuo.

Ambiente	Accesible	Determinista	Episódico	Estático	Discreto
Sistema de diagnóstico	No	No	No	No	No
Sistema de análisis de imágenes satelitales	Sí	Sí	Sí	Semi	No

Controlador de refinería	No	No	No	No	No
Tutor interactivo de inglés	No	No	No	No	Sí

Tabla 1.6 Ejemplos de tipos de ambiente (Russell 1996).

1.3.2. TIPOS DE AGENTES

Los agentes pueden clasificarse de acuerdo a diferentes parámetros. Una clasificación basada en el tipo de aplicación se muestra en la Tabla 1.7.

Tipo de Agente	Descripción
Agente de interfaz	Generan interfaces comprensibles para los usuarios en sistemas con alta carga de información.
Agentes de sistemas	En sistemas distribuidos complejos permiten realizar inventarios de hardware, interpretar eventos de red, manipular dispositivos de respaldo, detección de virus, entre otros.
Agentes de navegación	Se utilizan para navegar sobre sistemas conectados en red, al recordar sitios o direcciones de interés así como información relacionada.
Agentes de monitoreo	Proporcionan información de manera eficaz y oportuna cuando ocurre un evento.
Agentes consejeros	Proporcionan consejos a los usuarios en el uso de herramientas o en sistemas de diagnóstico o ayuda.
Agentes de recomendación	A medida que interactúa con el usuario va recomendando en base a información acerca de perfiles y preferencias de usuarios acerca de ítems o tópicos en particular.

Tabla 1.7 Clasificación de los agentes de acuerdo al tipo de aplicación (Jiménez 2000).

De acuerdo al tipo de percepciones y acciones los agentes se clasifican en:

- Agente de reflejo simple
- Agentes bien informados
- Agentes basados en metas
- Agentes basados en utilidad

En la Figura 1.11 se presenta la relación de estos agentes de acuerdo a su nivel de complejidad.

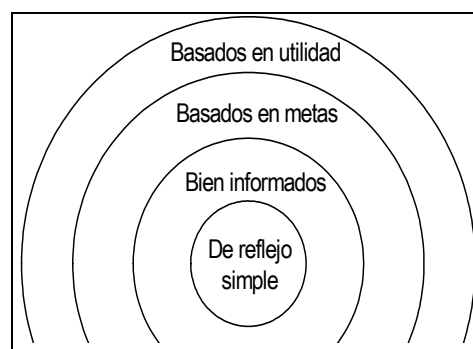


Figura 1.11 Tipos de agentes de acuerdo su nivel del complejidad (Los Autores).

1.3.2.1. Agente de reflejo simple

Basan su operación en un conjunto de reglas del tipo condición-acción que permiten establecer las conexiones entre las percepciones y las acciones. A través de los sensores el agente capta el estado actual del entorno, después busca en el conjunto de reglas una que coincida con lo percibido y qué acción debe ejecutarse. Este tipo de agente es fácil de implementar pero tiene limitaciones en el dominio de la aplicación.

1.3.2.2. Agentes bien informados

Tienen la capacidad de discernir entre estados del mundo generados por la misma entrada, sin embargo ejecutan acciones diferentes. Para esto se debe considerar el estado anterior del ambiente que al combinarlo con el conocimiento acerca de la manera cómo evoluciona el entorno y cómo las acciones del agente afectan al mundo. Se tiene el estado actual con el que el agente busca la regla correspondiente (condición-acción) y procede a ejecutarla.

1.3.2.3. Agentes basados en metas

Poseen información sobre las metas que se pretenden alcanzar y el estado que prevalece en el ambiente. El agente combina este conocimiento con el resultado de las posibles acciones y ejecuta la mejor acción para satisfacer las metas de forma eficiente y eficaz. Esta operación comprende procesos de selección sencilla, procesos de búsqueda complejos y una planificación de las metas.

1.3.2.4. Agentes basados en utilidad

Permiten distinguir entre estados felices e infelices. Mediante una medida de desempeño se puede establecer una comparación entre los diversos estados del entorno para poder encontrar el estado de felicidad del agente. Además, contiene información relacionada con las metas y la manera cómo

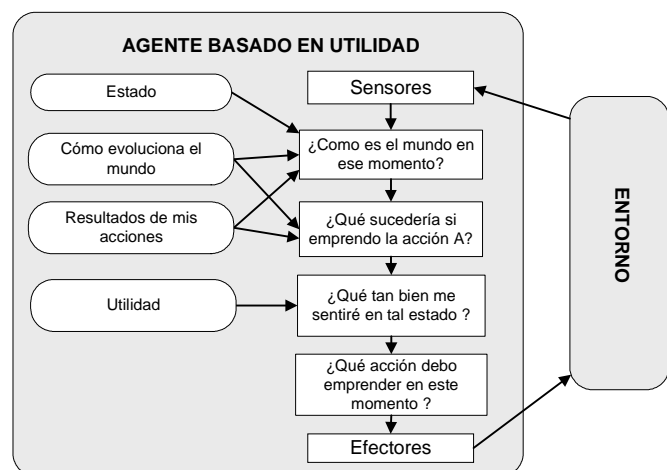


Figura 1.12 Agente basado en utilidad (Russell 1996).

evoluciona el mundo. Incluso conocen los resultados de sus acciones. En la Figura 1.12 se muestra un esquema.

1.3.3. AGENTES COMO MEDIADORES EN EL E-COMMERCE

El éxito de un comercio electrónico depende de varios factores: debe ser innovador, debe dar valor añadido a los productos y proveer información e interacción que otros sitios no poseen. Además, el negocio debe construir una comunidad de usuarios de manera que el sitio se convierta en una puerta de entrada para el desarrollo de actividades comerciales.

Los agentes inteligentes incorporan mejoras interesantes para el área del comercio electrónico y permiten conseguir de una manera más efectiva las metas del negocio. En la Tabla 1.8 se presentan los principales objetivos de un negocio electrónico junto con la técnica de e-commerce que los implementa.

Objetivo de un negocio electrónico	Técnica de e-commerce
Ayudar a los visitantes nuevos e infrecuentes	Listas de recomendación extensas
Construir credibilidad a través de comunidades	Comentarios y ratings de consumidores
Motivar a los clientes a regresar	Servicios de notificación
Realizar ventas cruzadas	Recomendaciones asociadas a productos
Construir relaciones a largo plazo	Personalización profunda

Tabla 1.8 Objetivos de un negocio electrónico y técnicas de e-commerce (Los Autores).

Un enfoque interesante para analizar el papel de los agentes en el comercio electrónico es en base al modelo del comportamiento del consumidor, especialmente en la parte que comprende las decisiones y acciones involucradas en el consumo de productos y servicios. Por esta razón, emergen los siguientes tipos

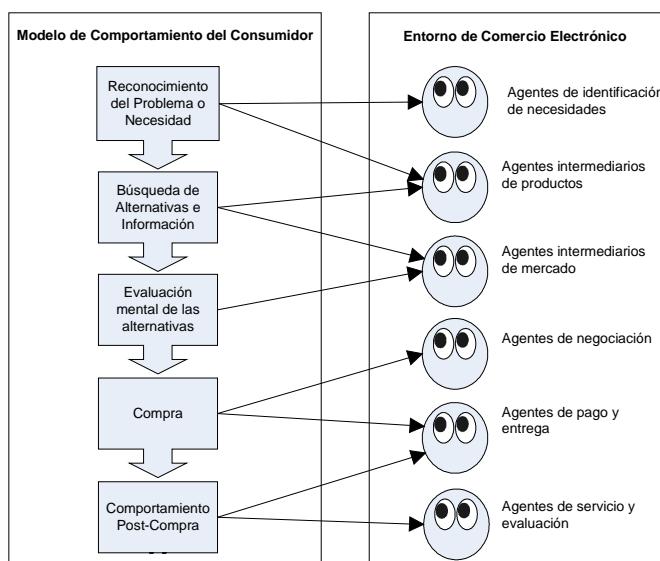


Figura 1.13 Agentes que soportan el modelo de comportamiento del consumidor (Los Autores).

de agentes (Figura 1.13):

- Agentes de identificación de necesidades
- Agentes de negociación
- Agentes intermediarios de productos
- Agentes de pago y entrega
- Agentes intermediarios de mercado
- Agentes de servicio y evaluación

1.3.3.1. Agentes de identificación de necesidades

Soportan la fase de reconocimiento del problema y su principal función es publicidad y promoción de productos o servicios. Presentan información con el objetivo de que usuario identifique cierto tipo de necesidades.

1.3.3.2. Agentes intermediarios de productos

Soportan las fases de reconocimiento del problema y búsqueda de alternativas. Ayudan al usuario a determinar qué comprar. Muestran las mejores alternativas basadas en criterios provistos por el usuario. Un ejemplo de agentes intermediarios son los agentes de recomendación de productos.

1.3.3.3. Agentes intermediarios de mercado

Soportan las fases de búsqueda y evaluación de alternativas. Ayudan al usuario a determinar dónde comprar. Consultan información de diferentes tiendas electrónicas para evaluar precios y características de productos.

1.3.3.4. Agentes de negociación

Soportan la fase de compra. Ayudan a determinar el precio de una transacción comercial. Son utilizados en sistemas de subasta en línea en donde el usuario configura ciertas reglas para la negociación.

1.3.3.5. Agentes de pago y entrega

Soportan las fases de compra y post-compra. Proveen de las seguridades del caso para completar la transacción comercial al mismo tiempo que muestran información de las alternativas y estado de entrega.

1.3.3.6. Agentes de servicio y evaluación

Soportan la fase de post-compra. Tienen el objetivo de evaluar los niveles de satisfacción e incrementar lo niveles de lealtad del cliente.

1.3.4. AGENTES DE RECOMENDACIÓN DE PRODUCTOS

Son agentes que implementan una parte de la estrategia de e-marketing del negocio (§ 1.1.2) y soportan las fases iniciales del modelo del comportamiento del consumidor. Este tipo de agentes de recomendación utilizan la información del mercado meta y varios elementos del marketing mix para presentar al usuario recomendaciones relacionadas con productos y servicios. En el Anexo I se presenta la correlación de las técnicas de e-commerce de la Tabla 1.8 y varios métodos de recomendación de uno de los sitios de comercio electrónico más populares sobre Internet.

La funcionalidad de estos agentes depende fundamentalmente de procesos de recomendación cuyas entradas y salidas corresponde a las percepciones y acciones dentro de un ambiente de comercio electrónico. El proceso de recomendación está conformado por los siguientes elementos (Figura 1.19): entradas y salidas funcionales, método de recomendación y adaptación de la recomendación.

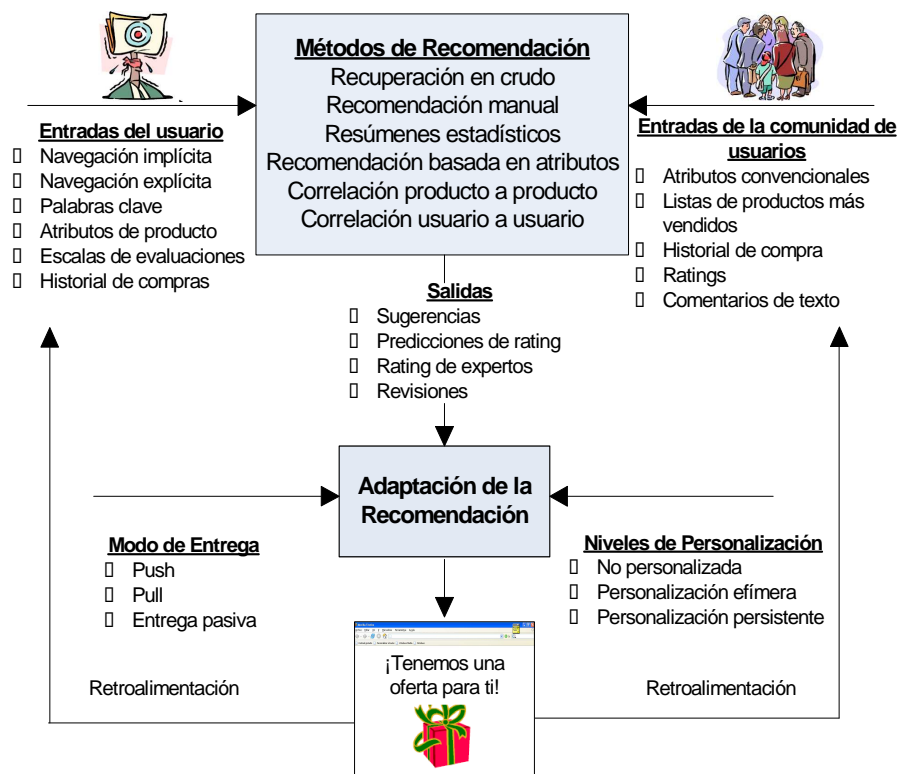


Figura 1.14 Proceso de recomendación (Shafer 2000).

1.3.4.1. Entradas y salidas funcionales

La información del mercado meta es obtenida a través de diversos métodos de

recolección de información del e-marketing. Las entradas y salidas constituyen el flujo del proceso de recomendación; además, todas sus variedades son opcionales y dependen del método de recomendación utilizado por el agente.

1.3.4.1.1. Entradas del usuario

Esta información proviene del usuario y permite la realización de tareas del análisis del consumidor; además, habilita a los agentes para que realicen recomendaciones con gran nivel de personalización. Los principales tipos de entradas de usuario son los siguientes:

- **Navegación implícita:** Son el conjunto de entradas tomadas directamente de la actividad del usuario quien no está conciente de que son utilizadas en el proceso de recomendación. Dentro de este tipo de entradas se encuentran las categorías y productos visitados así como también los productos ingresados al carro de compras.
- **Navegación explícita:** Son el conjunto de entradas ingresadas intencionalmente por el usuario para informar al sistema acerca de sus preferencias generales. Ej.: Formularios de perfil de usuario.
- **Palabras clave:** Son cadenas de texto ingresadas en las búsquedas.
- **Atributos de producto:** Son entradas en donde el cliente expresa sus necesidades e intereses a través de la especificación de los atributos deseados respecto a un producto o servicio.
- **Escalas de evaluación:** Son escalas cualitativas o cuantitativas que indican el grado de interés de un cliente respecto a un producto.
- **Historial de compras:** Permite determinar cuáles son los productos por los cuales el cliente ha expresado cierto tipo de interés en concreto.

1.3.4.1.2. Entradas de la comunidad de usuarios

Constituyen la síntesis de información ingresada por un grupo grande de consumidores y permiten obtener datos de segmentación de mercado. Las principales entradas de la comunidad de usuarios son:

- **Atributos convencionales:** Se refieren a atributos de producto ampliamente aceptados por una comunidad. Ej.: género de película o categoría de libro.
- **Listas de productos más vendidos:** Comprenden los productos o servicios

que han generado mayor volumen de ventas no solamente a nivel comercio electrónico sino a nivel de mercado en general.

- **Historial de compra:** Constituye el conjunto de registros de compras de una comunidad a partir de la cual se puede realizar minado de datos y obtener información estadística útil.
- **Ratings:** Constituyen los resúmenes estadísticos de las evaluaciones de productos ingresadas por los usuarios.
- **Comentarios de texto:** Varios sitios motivan a sus usuarios a escribir comentarios de productos para facilitar el proceso de decisión del consumidor.

1.3.4.1.3. Salidas

Son un conjunto de información que soporta las tres primeras fases del modelo del comportamiento del consumidor y están conformadas por un conjunto de recomendaciones cuyo contenido varían en tipo, cantidad y presentación. Dentro de los principales tipos de salidas se encuentran las siguientes:

- **Sugerencias:** Son el tipo más común de salida de los sistemas de recomendación. Solamente con poner a consideración del consumidor un producto se puede incrementar la probabilidad de compra. Es importante tomar en cuenta que una sugerencia puede ser rechazada porque el usuario ya posee el producto. Algunas estrategias de recomendación presentan los producto en manera desordenada mientras que otros presentan listas ordenadas alfabéticamente para evitar que se descarte un conjunto completo de recomendaciones.
- **Predicciones de rating:** Algunos algoritmos permiten dar estimaciones del rating que daría un usuario. Estas estimaciones pueden ser orientadas a usuarios individuales o comunidades y pueden ayudar a los consumidores a entender la importancia de la recomendación.
- **Rating de expertos:** Dentro de comunidades de usuarios pequeñas cuyos miembros son conocidos se pueden mostrar ratings individuales hechos por especialistas y críticos los cuales pueden ir acompañados por comentarios. Por ejemplo recomendaciones de películas o textos.
- **Revisiones:** Son tipos de recomendaciones que contienen información en forma de comentario de texto De esta manera se provee el usuario de una

herramienta de ayuda para la evaluación de cierto producto.

1.3.4.2. Método de recomendación

Se encarga de combinar los elementos precio y producto del marketing mix. Dentro de los principales métodos de recomendación se encuentran los siguientes:

- **Recuperación en crudo:** Consiste en un tipo de búsqueda simple en una base de datos de productos. La mayoría de aplicaciones de e-commerce poseen este tipo de búsqueda.
- **Recomendación manual:** Consiste en un grupo de recomendaciones elaboradas manualmente por especialistas como por ejemplo editores, artistas, críticos y otros expertos los cuales se basan en sus propios gustos e intereses. Estas recomendaciones son acompañadas por comentarios de texto para ayudar al consumidor a evaluar la recomendación.
- **Resúmenes estadísticos:** En alguno a casos en donde la personalización es innecesaria se provee al usuario de resúmenes estadísticos de la opinión de una comunidad. Ej,: porcentajes de compras o rating promedio de un producto.
- **Recomendación basada en atributos:** Este método devuelve productos que se ajustan mejor a los atributos o características solicitadas por el consumidor. Por ejemplo, un cliente solicita un computador con una lista de características técnicas específicas y el sistema le presenta una lista con las recomendaciones que más se ajustan la petición.
- **Correlación producto a producto:** Identifica relaciones entre productos basadas en los intereses del consumidor. Existen diferentes técnicas, sin embargo la mayoría asocian productos en base los intereses presentados en el momento de la navegación y no en el historial de compra.
- **Correlación usuario a usuario:** Este método de recomendación se basa en la relaciones de consumidores con perfil similar que han comprado productos en un sitio de e-commerce.

1.3.4.3. Adaptación de la recomendación

Los resultados generados por el proceso de recomendación pueden sufrir cierto grado de transformación para definir la manera cómo y cuándo van a ser

presentadas las recomendaciones con el uso de técnicas de e-marketing y e-CRM. Los procesos de adaptación combinan los elementos de comunicación y distribución del marketing mix. Las maneras de adaptación de las recomendaciones se dan en función del nivel de personalización y el modo de entrega.

1.3.4.3.1. Nivel de personalización

Los grados de personalización utilizan varias técnicas de e-CRM y corresponden a los siguientes niveles:

- **No personalizado:** Se da cuando se producen las mismas recomendaciones a todos los usuarios las cuales pueden ser basadas en selección manual, resúmenes estadísticos u otras técnicas.
- **Personalización efímera:** Las recomendaciones se basan en la navegación del usuario en el sitio. Generalmente se basa en correlación producto a producto o en recomendación basada en atributos.
- **Personalización persistente:** Se da cuando se producen diferentes recomendaciones para todos los usuarios. Para esto es necesario la configuración de perfiles de usuario y sus intereses.

1.3.4.3.2. Modo de entrega

Las maneras cómo el cliente puede obtener recomendaciones se basan en las técnicas de comunicación de e-marketing y son las siguientes:

- **Modelo Push:** se da cuando el sitio atrae al cliente. Por ejemplo, al ofrecer promociones vía mensajes de correo electrónico con vínculos hacia un sitio de e-commerce.
- **Modelo Pull:** Se da cuando el consumidor está buscando información. En este caso el usuario está conciente que las recomendaciones están disponibles al navegar un sitio de comercio electrónico; sin embargo, se muestran solamente cuando el cliente las solicita.
- **Entrega pasiva:** Se da cuando las recomendaciones se presentan en el contexto de la navegación. Generalmente se presentan ofertas relacionadas con el producto que está siendo mostrado actualmente.

CAPÍTULO 2. ANÁLISIS Y DISEÑO DEL AGENTE

2.1. METODOLOGÍA DE DESARROLLO

Con respecto a la ingeniería de software no se utilizará una metodología de desarrollo sino el Rational Unified Process (RUP) que es un framework de procesos y se describe en el Anexo IV.

2.2. ANÁLISIS

2.2.1. CONTEXTO DEL PROBLEMA

Debido a la dinámica y al constante cambio de los segmentos de mercado y tendencias de consumo dentro del ámbito de los negocios electrónicos, es necesario que el avance tecnológico soporte tales cambios a bajo costo y dé al cliente un mejor servicio que sea personalizado y amigable.

Por otro lado, los agentes inteligentes están enfocados a tener cierto grado de autonomía y flexibilidad respecto al ambiente en que se desenvuelven. Esta flexibilidad puede ser implementada con el uso de las ontologías informáticas, las cuales ayudan a expresar de una manera formal un conjunto de conceptos semánticamente relacionados dentro de un área determinada del conocimiento.

Por lo dicho anteriormente, los agentes inteligentes, en colaboración con las aplicaciones de negocios, pueden soportar con gran flexibilidad la dinámica y los cambios en dicho contexto teniendo como resultado soluciones más versátiles y eficientes.

2.2.2. DESCRIPCIÓN DEL PROBLEMA

La tecnología de e-commerce en el Ecuador está siendo cada vez más popular. Muchos negocios están adoptando la modalidad de vender sus productos y servicios a través de Internet. Mientras mayor número de sistemas de comercio electrónico aparezcan en la Web mayor será en nivel de competencia de sitios que vendan productos similares. Por esta razón, una de las opciones para incrementar el nivel de ventas es a través del uso de técnicas de recomendación.

Por otra parte, el desarrollo de un módulo de recomendación incrementa en gran parte el costo del desarrollo de un sistema de comercio electrónico. Además es necesario considerar que los sitios de comercio electrónico utilizan tecnologías

heterogéneas para su implementación. Incluso los diseños de sus bases de datos son diferentes.

Por lo dicho anteriormente, surge el problema de cómo implementar un agente de recomendación de productos o servicios de tipo genérico que sea capaz de comunicarse con cierto rango de aplicaciones de comercio electrónico y que estén implementadas con diferentes tecnologías de software y distintos modelos de bases de datos.

2.2.3. ALCANCE

El Agente de Recomendación de Productos o Servicios basado en Ontologías (ARPS) es un sistema de software que será capaz de interactuar dentro de un ambiente de comercio electrónico que funcione sobre Internet. El agente podrá ser adaptado a un negocio específico con el objetivo de brindar servicios de recomendación especializados, los cuales estarán en la capacidad de soportar tanto estrategias básicas de publicidad como también las etapas iniciales del proceso de compra del consumidor. Es necesario mencionar que el ARPS no incluye aspectos de seguridad ni herramientas de gestión de información acerca de las entidades y reglas del negocio como productos, usuarios, publicidad entre otros. El ARPS debe ser un sistema cuya configuración debe cubrir una amplia variedad de negocios electrónicos orientados a comercializar diferentes tipos de productos o servicios, por lo tanto debe ser capaz de interactuar con un sistema de e-commerce o un agente de software. Sin embargo, para lograr un nivel de interacción con dichos sistemas es necesario que éstos utilicen el mismo protocolo de comunicación del ARPS.

Por otro lado, para el funcionamiento del producto es necesario seguir un proceso de adaptación en base a la información del negocio específico al que va a brindar sus servicios de recomendación.

El proceso de adaptación del ARPS para un negocio específico está conformado por los siguientes pasos:

1. Configuración de la base de conocimiento del ARPS en base el negocio que va a soportar.
2. Migración y transformación de la información de la base de datos del negocio a la base de conocimiento del ARPS.

2.2.4. VISIÓN GENERAL DE LA APLICACIÓN

La aplicación que se va a desarrollar es un agente de recomendación que ayude a implementar de mejor manera estrategias básicas de e-marketing al mismo tiempo que incremente la fidelidad del cliente hacia sitio de comercio electrónico al que brinda sus servicios. Específicamente, el Agente debe ser capaz de soportar las dos primeras fases del proceso de compra del modelo de comportamiento del consumidor que son: reconocimiento del problema o necesidad y búsqueda de alternativas e información. Estas dos fases concuerdan con algunos objetivos de un negocio electrónico tomados de la Tabla 1.8 y se resumen en la Tabla 2.1. En esta tabla también se listan las técnicas de e-commerce que implementan estos objetivos y para cada técnica se define un aspecto de recomendación que se delega al Agente.

Objetivo del negocio	Técnica de e-commerce	Aspecto de recomendación
Ayudar a los visitantes nuevos e infrecuentes.	Listas de recomendación extensas.	Recomendación relacionada con partes y accesorios de productos.
Realizar ventas cruzadas.	Recomendaciones asociadas a productos.	Recomendaciones relacionadas con categorías de productos.
Construir relaciones a largo plazo.	Personalización profunda.	Recomendaciones en base al perfil de usuario (Recomendaciones de interés).

Tabla 2.1 Relación de objetivos del negocio electrónico, técnicas de e-commerce y aspectos de recomendación (Los Autores).

Por otra parte, la estrategia de marketing ayuda a cumplir en parte dichos objetivos del negocio electrónico por tanto en base a algunos elementos que intervienen en el marketing mix se definen los elementos de recomendación que se listan en la siguiente tabla:

Elemento de recomendación	Descripción/Explicación
Productos	Presentan opciones para la fase de búsqueda de alternativas al mismo tiempo que sirven para identificar necesidades.
Promociones	Combinan los elementos de precio y producto del marketing mix.
Publicidad	Es el elemento por excelencia para la identificación de necesidades por parte del cliente.
Servicios	Tiene la misma funcionalidad que productos con la diferencia de que agrupa solamente características intangibles.

Tabla 2.2 Elementos de recomendación (Los Autores).

2.2.5. ESPECIFICACIÓN DE REQUERIMIENTOS

Los requerimientos funcionales del ARPS (Agente de Recomendación de Productos o Servicios basado en Ontologías) proceden de la combinación de los aspectos de recomendación de la Tabla 2.1 y los elementos de recomendación de la Tabla 2.2; que además se listan a continuación:

- **Recomendación de productos:** En base a un producto o a un perfil de usuario, el agente busca los accesorios y partes de los productos, o los productos de interés asociados al perfil y los retorna en forma de lista de los productos.
- **Recomendación de servicios:** En base a un producto, categoría de productos o a un perfil de usuario, el agente busca los servicios asociados respectivamente y los presenta en forma de lista.
- **Presentación de promociones:** En base a un producto, categoría de productos o a un perfil de usuario el agente busca las promociones relacionadas respectivamente y las presenta en forma de lista.
- **Presentación de publicidad:** En base a un producto, categoría de productos o a un perfil de usuario el agente busca la publicidad asociada respectivamente y los presenta en forma de lista.

El Anexo II contiene el documento de Especificación de Requerimientos del Software ERS, el cual está basado en el estándar IEEE Std. 830-1998. En este documento se definen detalladamente los requerimientos funcionales y no funcionales del Agente así como también otros aspectos de la aplicación.

2.2.6. MODELO DE CASOS DE USO

En base al ERS (Anexo II) se determinaron los casos de uso de la Tabla 2.4 y dos tipos de actores: Cliente y Cliente Registrado (Tabla 2.3). El caso de uso principal es Solicitar Recomendación del cual se derivan los casos de uso de la Tabla 2.5 (Ver Figura 2.2).

Las relaciones entre actores y casos de uso se muestran en el diagrama de la Figura 2.1. Algunas de estas relaciones utilizan los estereotipos descritos en la Tabla 2.7. En la Tabla 2.6 se describen los paquetes de casos de uso agrupados.

2.2.6.1. Descripción de actores y casos de uso

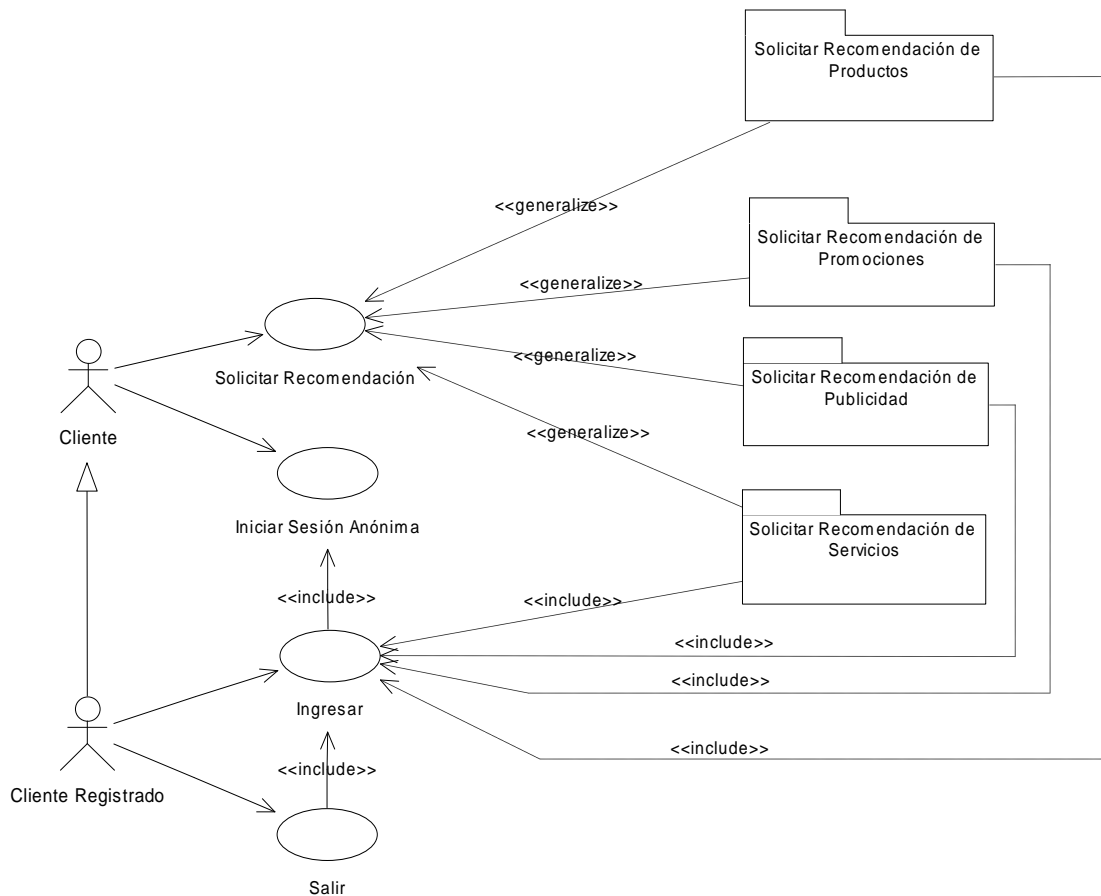


Figura 2.1 Diagrama de casos de uso (Los Autores).

Actor	Descripción
Cliente	Es una aplicación e-commerce u otro agente de software que utiliza el servicio de recomendación proporcionado por el ARPS (Agente de Recomendación de Productos o Servicios) a través de la Web.
Cliente Registrado	Es un Cliente que tiene un usuario con perfil registrado.

Tabla 2.3 Descripción de actores (Los Autores)

Caso de uso	Descripción
Solicitar Recomendación	Permite al Cliente consultar productos, promociones, publicidad o servicios relacionados a un producto en particular o una categoría de productos.
Iniciar Sesión Anónima	Permite que el Cliente ingrese un código de sesión anónima en el contexto del Agente.
Ingresar	Permite al Cliente identificarse con usuario y contraseña de tal forma que el Agente pueda hacer uso de su perfil.
Salir	Permite el Cliente terminar la sesión con el Agente.

Solicitar Recomendación de Productos	Permite al Cliente consultar partes y accesorios de un producto en particular además de productos con alguna característica especial que pertenecen a una Categoría. El Cliente Registrado puede consultar productos que pueden ser de interés de acuerdo a su perfil.
Solicitar Recomendación de Promociones	Permite al Cliente consultar promociones relacionadas a una categoría o a partes y accesorios de un producto en particular. El Cliente Registrado puede consultar promociones que pueden ser de interés de acuerdo a su perfil.
Solicitar Recomendación de Publicidad	Permite al Cliente consultar publicidad relacionada a una categoría o a partes y accesorios de un producto en particular. El Cliente Registrado puede consultar publicidad que puede ser de interés de acuerdo a su perfil.
Solicitar Recomendación de Servicios	Permite al Cliente consultar servicios relacionados a una categoría o partes y accesorios de un producto en particular. El Cliente Registrado puede consultar servicios que pueden ser de interés de acuerdo a su perfil.

Tabla 2.4 Descripción de casos de uso (Los Autores).

Casos de Uso		Aspecto de recomendación		
		Partes o Accesorios	Categorías	Recomendación de Interés
Elemento de Recomendación	Productos	Solicitar Partes o Accesorios	Solicitar Productos de la Categoría	Solicitar Productos de Interés
	Promociones	Solicitar Promociones de Partes o Accesorios	Solicitar Promociones de la Categoría	Solicitar Promociones de Interés
	Publicidad	Solicitar Publicidad de Partes o Accesorios	Solicitar Publicidad de la Categoría	Solicitar Publicidad de Interés
	Servicios	Solicitar Servicios de Partes o Accesorios	Solicitar Servicios de la Categoría	Solicitar Servicios de Interés

Tabla 2.5 Casos de uso resultantes de la combinación de los elementos de recomendación y tres aspectos relacionados a la recomendación (Los Autores).

Paquete de casos de uso	Descripción
Solicitar Recomendación de Productos	Agrupar actores y casos de uso relacionados con recomendación de productos.
Solicitar Recomendación de Promociones	Agrupar actores y casos de uso relacionados con recomendación de promociones.
Solicitar Recomendación de Publicidad	Agrupar actores y casos de uso relacionados con recomendación de publicidad.
Solicitar Recomendación de Servicios	Agrupar actores y casos de uso relacionados con recomendación de servicios.

Tabla 2.6 Descripción de paquetes de casos de uso (Los Autores).

Estereotipo	Descripción
<<include>>	A incluye B. Significa que para realizar el caso de uso A es necesario realizar al caso de uso B.
<<generalize>>	A generaliza B: Significa que el caso de uso B es una generalización del paquete de casos de uso A.

Tabla 2.7 Descripción de estereotipos (Los Autores).



Figura 2.2 Taxonomía del caso de uso Solicitar Recomendación (Los Autores).

2.2.6.2. Especificación de casos de uso

Los casos de uso son especificados en base de los paquetes de la Tabla 2.6 y se encuentran en el Anexo III. Además, se adjunta un diagrama de casos específico del paquete.

2.2.7. ARQUITECTURA DEL SOFTWARE A NIVEL DE ANÁLISIS

2.2.7.1. Representación Arquitectónica

El ARPS (Agente de Recomendación de Productos o Servicios basado en Ontologías) es una aplicación que provee servicios de recomendación a una aplicación de e-commerce externa la cual cuenta con su respectiva base de datos. Por otro lado, el Agente funciona como un servicio Web levantado sobre un servidor de aplicaciones el cual administra la concurrencia de peticiones por parte de la Aplicación Cliente. Además, cuenta con su propia base de conocimiento cuya información se almacena en archivos de ontologías (Figura 2.3). Además, para el análisis de esta arquitectura son necesarias las vistas de la Tabla 2.8 cuyos elementos son desarrollados en las secciones subsiguientes.

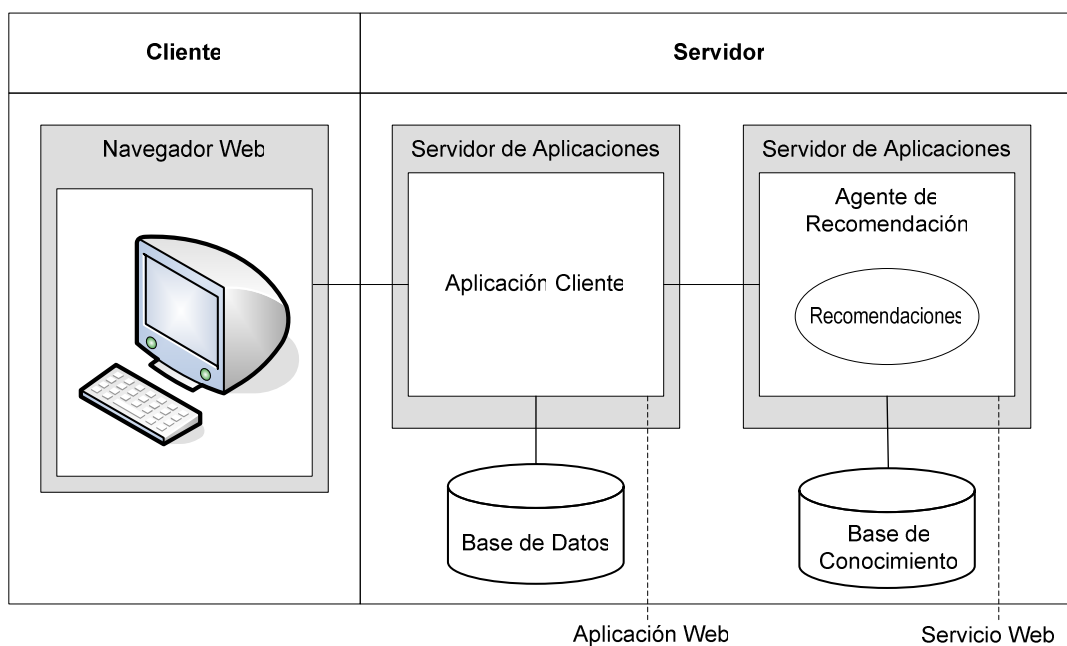


Figura 2.3 Descripción general de la arquitectura del ARPS (Los Autores).

Vista arquitectónica	Descripción
Vista de requerimientos	Contiene un listado de varios mecanismos de análisis.
Vista lógica	Describe el modelo de diseño a nivel de análisis y contiene una realización de casos de uso.
Vista de distribución	Describe el modelo de distribución inicial del sistema a nivel de nodos físicos.

Tabla 2.8 Vistas arquitectónicas a nivel de análisis (Los Autores).

2.2.7.2. Vista de requerimientos: Mecanismos de análisis

En base a la sección 3.3 del ERS (Anexo II) surgen varios aspectos del sistema que deben ser tomados en cuenta para la arquitectura del software por lo que se determinan los siguientes mecanismos de análisis (Tabla 2.9):

Nombre	Descripción
Tipo de sistema de información	Se refiere a la modalidad que se va a utilizar para coleccionar, procesar y transmitir la información organizadamente conforme a un proceso predefinido sea manual o automático.
Técnica de representación del conocimiento	Se refiere a la notación o formalismo que se va a utilizar para representar la información o conocimiento del dominio del problema.
Método de inferencia	Se refiere al algoritmo que se va a utilizar con la base de conocimiento para realizar inferencias.
Procesamiento de peticiones del usuario	Controla las peticiones de al sistema hechas por una aplicación cliente.
Almacenamiento de la información	Es el sistema de almacenamiento persistente de la información.
Administración de parámetros	Se refiere al manejo de los parámetros necesarios para la configuración del sistema.
Autenticación	Verifica que el Cliente tiene las credenciales para acceder al sistema.

Tabla 2.9 Mecanismos de análisis (Los Autores).

2.2.7.3. Vista lógica: Modelo de diseño

Ver Anexo III.

2.2.7.4. Vista de distribución: Modelo de distribución inicial

Ver Anexo III.

2.3. DISEÑO

2.3.1. ARQUITECTURA DEL SOFTWARE A NIVEL DE DISEÑO

2.3.1.1. Vista de Requerimientos: Mecanismos de diseño e implementación

En base a los mecanismos de análisis descritos en la Tabla 2.9, se identifican los siguientes mecanismos de diseño e implementación:

Mecanismos de análisis	Mecanismos de diseño	Mecanismos de implementación
Tipo de sistema de información	-Arquitectura de sistema orientado a objetos. -Arquitectura de sistema basado en conocimiento. -Arquitectura de agente inteligente.	-El sistema cuenta con componentes con sus respectivas interfaces. -El sistema tiene base de conocimiento, base de hechos, motor de inferencia y unidades de entrada y salida. -El agente tiene un PAMA.
Técnica de representación del conocimiento	El sistema utiliza una base de conocimiento cuya información servirá para realizar inferencias	-El conocimiento se representa con lógica de predicados. -Se representa con reglas de producción. -Se representa con objetos. -Se representa con ontologías.
Método de inferencia	El sistema carga la información de la base de conocimiento y aplica un algoritmo para obtener realizar inferencia sobre la base de conocimiento.	-Las inferencias se realizan con el método de razonamiento progresivo. -Las inferencias se realizan con el método de razonamiento regresivo. -Las inferencias se realizan utilizando un método mixto.
Procesamiento de peticiones del usuario	Patrón de diseño controlador frontal (Front Controller).	- Aplicación servidor. - Servicio Web.
Almacenamiento de información	Modalidad de almacenamiento persistente para la información de la aplicación.	- La información se almacena en una base de datos. - La información se almacena en archivos de texto.
Administración de parámetros	Los parámetros se guardan externamente y son leídos por el sistema.	-Los parámetros son almacenados en un archivo XML. -Los parámetros son almacenados en un BDD
Autenticación	Basado en una contraseña y un identificador de usuario.	El identificador de usuario y su contraseña son autenticados contra la base de conocimiento.

Tabla 2.10 Mecanismos de diseño e implementación (Los Autores).

Tipo de sistemas de información	Ventajas	Desventajas
Sistema orientado a objetos	<ul style="list-style-type: none"> -Existe una fuerte unidad entre procedimientos (métodos) y datos. -Se puede tener un alto grado de modularización. -Mayor facilidad de trabajar con sistemas grandes. 	<ul style="list-style-type: none"> -Conocimiento y procesamiento combinados en un programa. -Los procesos no pueden contener errores. -Los cambios tardan más tiempo. -Necesita información completa para operar.
Sistema basado en conocimiento	<ul style="list-style-type: none"> -Base de conocimiento separada del mecanismo de procesamiento. -Los cambios en las reglas son fáciles. -El sistema puede funcionar con pocas reglas. 	<ul style="list-style-type: none"> -La base de conocimiento no tiene un alto nivel de modularización. -No tiene módulo de explicación.
Agente inteligente	<ul style="list-style-type: none"> - Constituyen una técnica para desarrollar software que ayuda a administrar la complejidad inherente al software. -Permiten modelar sistemas como una colección de agentes autónomos interactuando entre si. -Esta tecnología es más apropiada para construir sistemas en los que los datos, control o recursos están distribuidos. - Dan una metáfora natural para dar funcionalidad al sistema. -Los agentes inteligentes liberan de la tarea de discriminar entre un conjunto de información que se tiene acerca de un tema para poder obtener la información que sea más conveniente para su uso. 	<ul style="list-style-type: none"> -A pesar de que existen ya numerosas aplicaciones basadas en agentes, todavía no es una tecnología universal, y dada su inmadurez tiene claras desventajas frente a otros paradigmas, al momento de seleccionar una metodología. -Muchos de los sistemas que se han construido basados en agentes se podrían haber escritos fácilmente utilizando técnicas no agentes.

Tabla 2.11 Ventajas y desventajas de los tipos de sistemas de información.

Representación del conocimiento	Ventajas	Desventajas
Lógica de predicados	<ul style="list-style-type: none"> -La inferencia a partir de premisas verdaderas garantiza conclusiones verdaderas. -A partir de hechos particulares se puede tener conclusiones generales. 	<ul style="list-style-type: none"> -No maneja grados de incertidumbre. -Es difícil trabajar con hechos que cambian su valor de verdad con el tiempo.

Reglas de producción	<ul style="list-style-type: none"> -Es la forma más común de representar el conocimiento, debido a su gran sencillez. -Permiten el descubrimiento de nuevos hechos. -Los hechos y afirmaciones pueden ser retractados en el momento en que dejen de ser verdaderos. -Se puede aceptar incertidumbre en el proceso de razonamiento. 	<ul style="list-style-type: none"> -El crecimiento muy rápido número de reglas hace lento al proceso de inferencia y conduce a la introducción repeticiones o contradicciones.
Ontologías	<ul style="list-style-type: none"> - Sintetiza las mejores características de otras técnicas. -Permiten un alto grado de abstracción para resolver problemas. -El conocimiento es significativamente más estructurado y organizado. -La base de conocimiento puede ser modularizada. -La base de conocimiento es mantenible. -Las bases de conocimiento pueden ser reutilizadas. 	<ul style="list-style-type: none"> -Falta de estandarización de ontologías de alto nivel. -Dificultades de la validación por parte de ingenieros del dominio. -Es una técnica que está en proceso de investigación y no ha madurado aún.

Tabla 2.12 Ventajas y desventajas de las técnicas de representación del conocimiento.

Método de inferencia	Ventajas	Desventajas
Razonamiento progresivo	<ul style="list-style-type: none"> -Inicia con los hechos disponibles y usa reglas de inferencias para extraer más hechos. -Si se generan más hechos se disparan más reglas de inferencia. 	<ul style="list-style-type: none"> -El proceso completo no es guiado por objetivo así que no se puede saber cuando detener la aplicación de reglas.
Razonamiento regresivo	<ul style="list-style-type: none"> -Trabaja con una lista de metas o hipótesis y trata de demostrar si hay datos disponibles que soporten estas hipótesis. -La búsqueda es guiada por objetivo de esta manera solamente se aplican las reglas que son necesarias para llegar a la meta. 	<ul style="list-style-type: none"> -La meta debe ser conocida. -Muchos sistemas de inteligencia artificial pueden ser formulados en la modalidad de una búsqueda guiada por objetivo. -Este tipo de razonamiento es limitado en comparación con el razonamiento progresivo.
Razonamiento híbrido	<ul style="list-style-type: none"> -Se tiene las ventajas del razonamiento progresivo y regresivo. 	<ul style="list-style-type: none"> --Las desventajas entre los dos tipos de razonamiento se compensan.

Tabla 2.13 Ventajas y desventajas de los métodos de inferencia.

Procesamiento de peticiones del usuario	Ventajas	Desventajas
Aplicación servidor	-Es más eficiente que un servicio Web.	-No se cuenta con un protocolo estándar de comunicaciones.
Servicio Web	-Aportan interoperabilidad entre aplicaciones independientemente de las plataformas. -Utilizan protocolos estandarizados.	-Su rendimiento es bajo.

Tabla 2.14 Ventajas y desventajas del tipo de procesamiento de peticiones de usuario.

Una vez identificados los mecanismos de diseño e implementación, se procede a seleccionar los más convenientes y se describen en la Tabla 2.15.

Mecanismo	Selección / Justificación
Tipo de sistema de información	Se escoge el paradigma orientado a agentes porque provee una metáfora natural para representar la dinámica que se da en un ambiente de comercio electrónico.
Técnica de representación del conocimiento	Se escoge ontologías debido a la flexibilidad que tiene su estructura para organizar el conocimiento de expertos en este caso en el campo del comercio electrónico y el e-marketing. Además es una técnica que agrupa muchos beneficios de otras técnicas de representación del conocimiento a adicionalmente permite hacer inferencias sobre la base de conocimiento.
Método de inferencia	Se escoge un tipo de razonamiento híbrido porque se necesita razonamiento regresivo para verificar si una categoría de productos pertenece a la línea de descendencia de otra categoría. Por otro lado, es necesario correr reglas sobre hechos que describen ciertos eventos de usuario.
Procesamiento de peticiones del usuario	Se adoptará el patrón de diseño de J2EE controlador frontal cuya principal función es despachar las peticiones de otra aplicación. Esta función se implementa mejor como un servicio Web.
Almacenamiento de información	La información se almacenará en archivos de texto debido a que toda la base de conocimiento se carga en memoria lo cual no afecta el desempeño.
Administración de parámetros	Los parámetros se almacenarán en un archivo XML debido a que es un formato estándar basado en archivos de textos que facilita la estructuración de la información.

Tabla 2.15 Selección de mecanismos (Los Autores).

2.3.1.2. Diseño del Agente de Recomendación

En base a la fase de análisis se define el PAMA (percepciones, acciones, metas, ambiente) del Agente de Recomendación que se presenta en la Tabla 2.16. Las metas del Agente se basan en el soporte de las dos primeras fases del proceso de compra del modelo de comportamiento del consumidor (§1.1.4.2) y es preciso mencionar que dichas metas no necesariamente pueden estar representadas en la base de conocimiento. Por otro lado, el Agente se desenvuelve en un ambiente de comercio electrónico cuyas características describen en la Tabla 2.17. Además, en base al PAMA y al ambiente se determina que el ARPS es un agente de reflejo simple (§ 1.3.2.1).

PAMA	Descripción
Percepciones	<ul style="list-style-type: none"> • Productos • Categorías de productos • Usuarios • Credenciales de usuarios
Acciones	<ul style="list-style-type: none"> • Recomendación de productos • Recomendación de promociones • Recomendación de publicidad • Recomendación de servicios • Identificación de clientes • Recomendación personalizada
Metas	<ul style="list-style-type: none"> • Ayudar a la identificación de necesidades del cliente. • Presentar alternativas que incentiven la decisión de compra del cliente.
Ambiente	<ul style="list-style-type: none"> • Comercio electrónico

Tabla 2.16 PAMA del Agente de Recomendación (Los Autores).

Tipo de Ambiente	Explicación/Justificación
Inaccesible	El Agente no cuenta con el historial de acciones propias, acciones del cliente ni compras pasadas.
No determinista	El Agente no es capaz de determinar si el cliente compró un producto recomendado o si hizo clic en algún elemento recomendado.
No episódico	Las recomendaciones personalizadas del Agente dependen de que el usuario se haya autenticado con anterioridad.
Estático	El Agente no necesita observar el ambiente mientras decide sobre el curso de una acción.
Discreto	Todas las acciones y percepciones del Agente están perfectamente identificadas.

Tabla 2.17 Descripción del ambiente de comercio electrónico del ARPS (Los Autores).

2.3.1.3. Vista lógica: Modelo de diseño

En este modelo se definen siguientes estereotipos de paquetes:

Estereotipo	Descripción
<<intelligent agent>>	Empaqueta los elementos de un agente inteligente.
<<ontology>>	Empaqueta los elementos de una ontología.
<<web service>>	Empaqueta a un servicio Web.
<<sensors>>	Empaqueta los sensores de un agente inteligente.
<<efectors>>	Empaqueta los efectores de un agente inteligente.

Tabla 2.18 Estereotipos de paquetes del modelo de diseño (Los Autores).

Los estereotipos de métodos son descritos en la Tabla 2.19:

Estereotipo	Descripción
<<constructor>>	Constructor de objetos.
<<getter>>	Método para obtener el valor de un atributo.
<<setter>>	Método para ingresar el valor de un atributo.

Tabla 2.19 Estereotipos de métodos (Los Autores).

En base la realización de los casos de uso de la disciplina de análisis se clasifican las responsabilidades del Agente de Recomendación en dos grupos que se presentan en la Tabla 2.20.

Grupo	Responsabilidades
Responsabilidades asociadas al usuario	<ul style="list-style-type: none"> • Iniciar sesión. • Ingresar del sistema. • Salir del sistema.
Responsabilidades asociadas a la recomendación	<ul style="list-style-type: none"> • Obtener recomendación de productos. • Obtener recomendación de servicios. • Obtener recomendación de promociones. • Obtener recomendación de publicidad.

Tabla 2.20 Responsabilidades del Agente de Recomendación (Los Autores).

2.3.1.3.1. Clases de diseño

En la Tabla 2.21 se describen los elementos que conforman el paquete RecommendationAgent (Figura 2.4):

Nombre	Descripción
IRecommendationAgent	Interfaz de Agente de Recomendación basada en los grupos de responsabilidades de la Tabla 2.20.
RecommendationAgent	Agente de Recomendación. Recibe percepciones y ejecuta acciones
RecommendationManager	Ejecuta las tareas de recomendación.
UserManager	Ejecuta tareas de gestión de usuarios.
InferenceEngine	Encapsula un motor de inferencia basado en reglas.
KnowledgeBase	Encapsula la base de conocimientos.
AgentMetadata	Encapsula constantes y parámetros del sistema.
RecommendationService	Representa un Servicio Web con las funciones de recomendación.
AgentSensors	Paquete de sensores.
AgentEfectors	Paquete de efectores.

Tabla 2.21 Elementos del paquete RecommendationAgent (Los Autores).

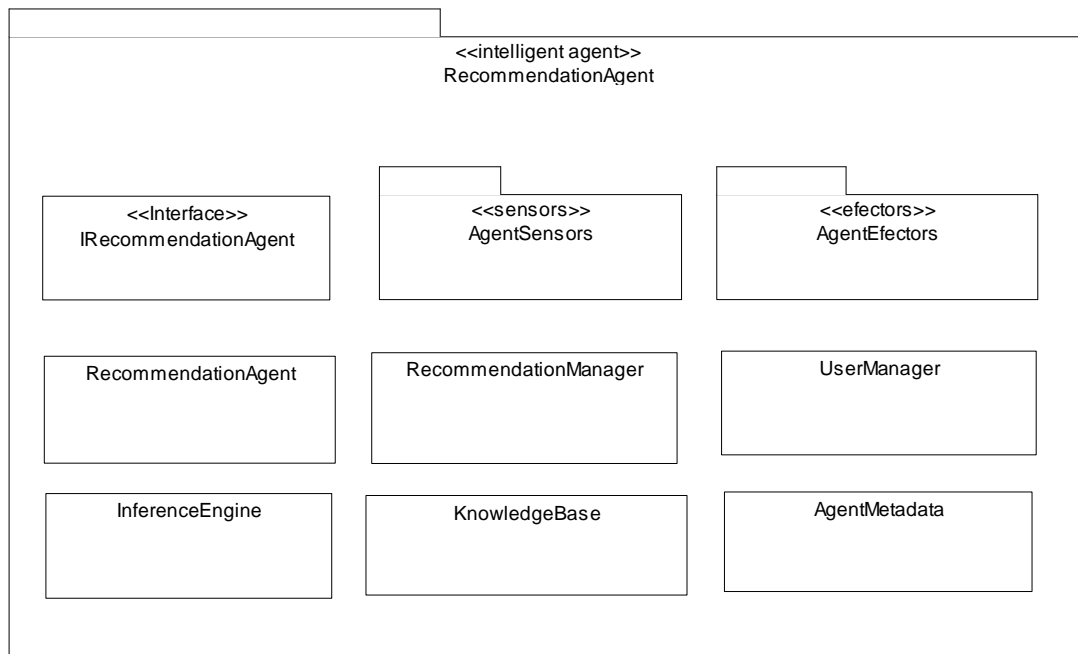


Figura 2.4 Paquete RecommendationAgent (Los Autores).

En base a la clasificación de responsabilidades de la Tabla 2.20 se ha creado el gestor de recomendaciones representado por la clase RecommendationManager y el gestor de usuarios representado por la clase UserManager, los cuales llevan a cabo las respectivas tareas del Agente y se pueden apreciar en el diagrama de la Figura 2.5.

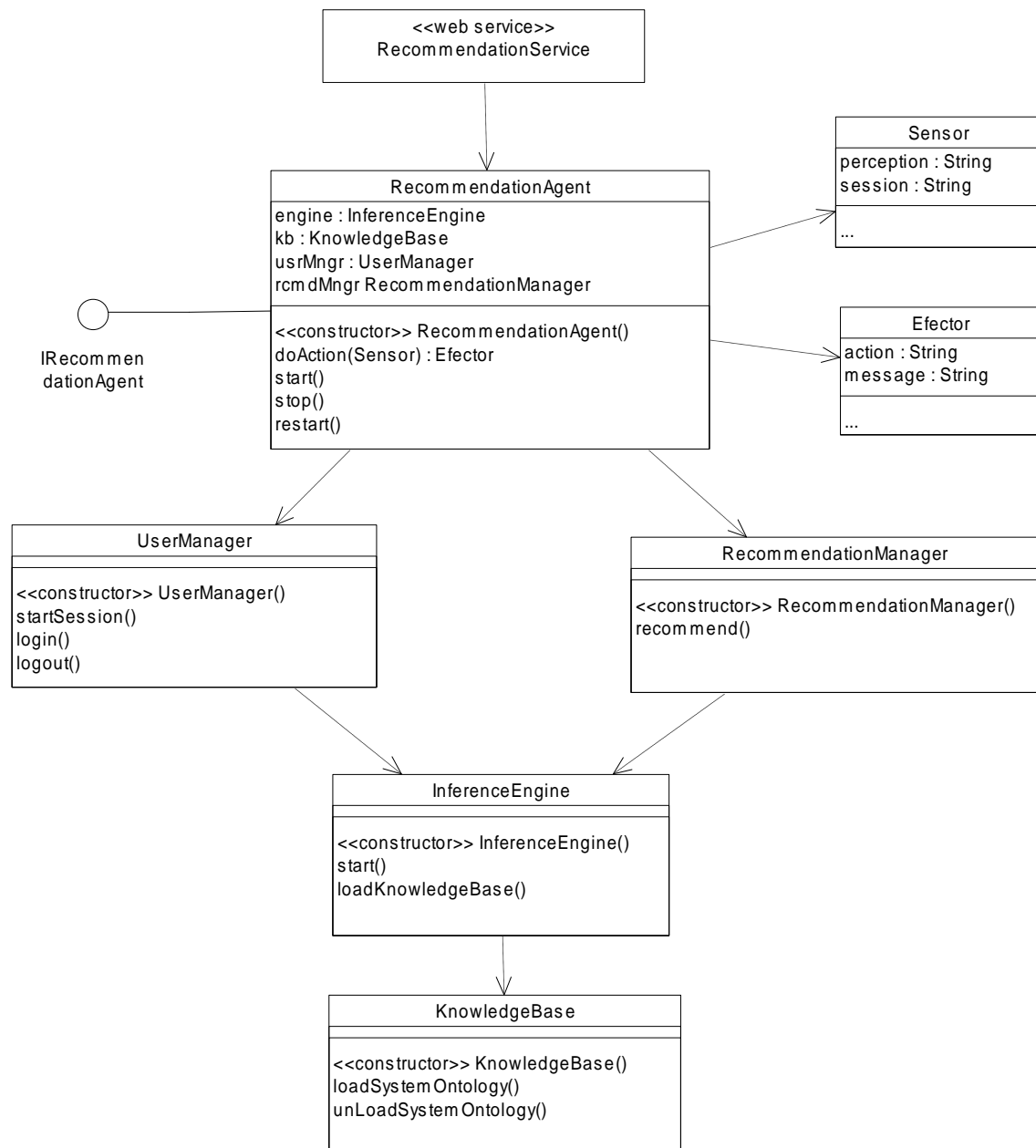


Figura 2.5 Diagrama de clases del Agente de Recomendación (Los Autores).

En la Tabla 2.22 presenta la descripción de las propiedades de las clases de los paquetes RecommendationAgent, AgentSensors y AgentEfectors.

En la Tabla 2.23 se describen los métodos de las clases del paquete RecommendationAgent y en la Tabla 2.24 se describen los métodos de la clase RecommendationService:

Clase	Propiedad	Descripción
RecommendationAgent	engine	Motor de inferencia.
	kb	Base de conocimiento.
	rcmdMngr	Gestor de recomendaciones.
	usrMngr	Gestor de usuarios.
Sensor	perception	Etiqueta que indica el tipo de percepción.
	session	Identificador de la sesión de usuario en la aplicación cliente.
RecommendationSensor	productCode	Código de producto.
	categoryCode	Código de categoría.
UserSensor	username	Nombre de usuario.
	password	Contraseña.
Efector	action	Identificador de la acción que ha ejecutado el agente.
	message	Mensaje de respuesta.
RecommendationEfector	recommendationList	Lista de recomendaciones.

Tabla 2.22 Propiedades de las clases de diseño (Los Autores).

Clase	Método	Descripción
RecommendationAgent	doAction	Recibir una percepción y ejecutar una acción.
	start	Iniciar Agente.
	stop	Detener el funcionamiento Agente.
	restart	Reiniciar Agente.
RecommendationManager	recommend	Realizar recomendación.
UserManager	startSession	Iniciar sesión anónima de un usuario en el Agente.
	login	Autenticar un usuario del Agente.
	logout	Cerrar sesión de usuario en el Agente.

InferenceEngine	start	Iniciar motor de inferencia.
	loadKnowledgebase	Cargar base de conocimiento.
KnowledgeBase	loadSystemOntology	Cargar la ontología del Agente.
	unloadSystemOntology	Descargar la ontología del Agente.

Tabla 2.23 Métodos de las clases de diseño (Los Autores)

Métodos	Descripción
productRecommend	Devuelve una lista de productos.
promotionRecommend	Devuelve una lista de promociones.
adRecommend	Devuelve una lista de publicidad.
serviceRecommend	Devuelve una lista de servicios.

Tabla 2.24 Métodos del Servicio Recomendación (Los Autores)

En la Figura 2.6 se puede observar el modelo de sensores y efectores del Agente y en la Tabla 2.25 se describen las clases de los paquetes AgentSensors y AgentEfectors:

Nombre	Descripción
Sensor	Sensor de agente inteligente.
RecommendationSensor	Encapsula percepciones relacionadas con tareas de recomendación.
UserSensor	Encapsula percepciones relacionadas con tareas de usuario.
Efactor	Efactor de agente inteligente. Encapsula los resultados de las tareas del agente.
RecommendationEfactor	Encapsula los resultados de las tareas de recomendación.

Tabla 2.25 Clases de los paquetes AgentSensors y AgentEfectors (Los Autores)

El la Figura 2.7 se presenta un diagrama de secuencia con el funcionamiento del Agente, en donde el servicio Web crea un objeto de tipo Sensor el cual sirve como parámetro del método doAction. Dependiendo de la percepción especificada en el Sensor se llama al gestor de usuarios (userManager) o al gestor de recomendaciones (RecommendationManager). Estos se comunican con el motor de inferencia y reciben los resultados de la percepción los cuales se los retorna a través de un objeto de clase Efactor.

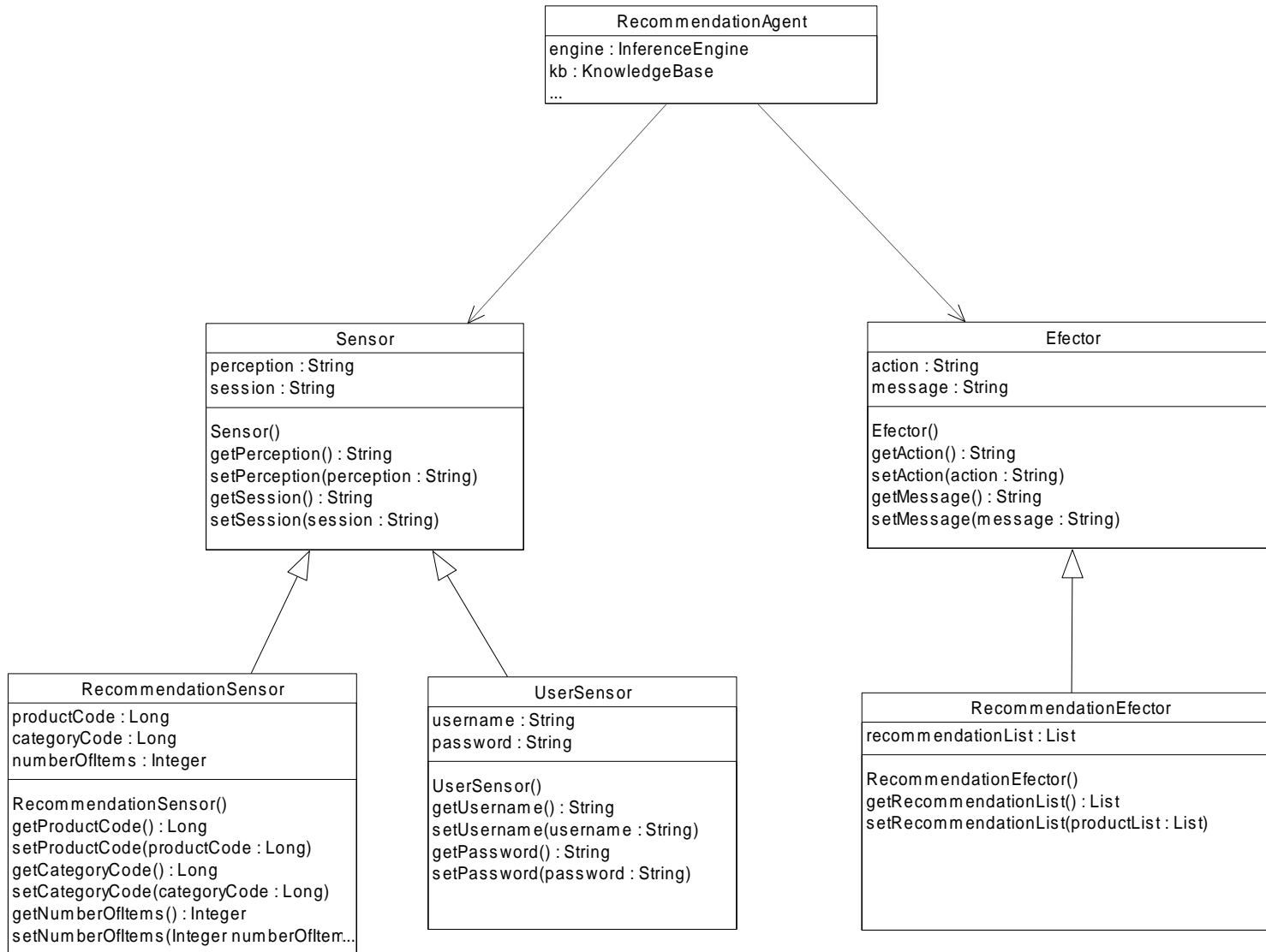


Figura 2.6 Diagrama de sensores y efectores (Los Autores).

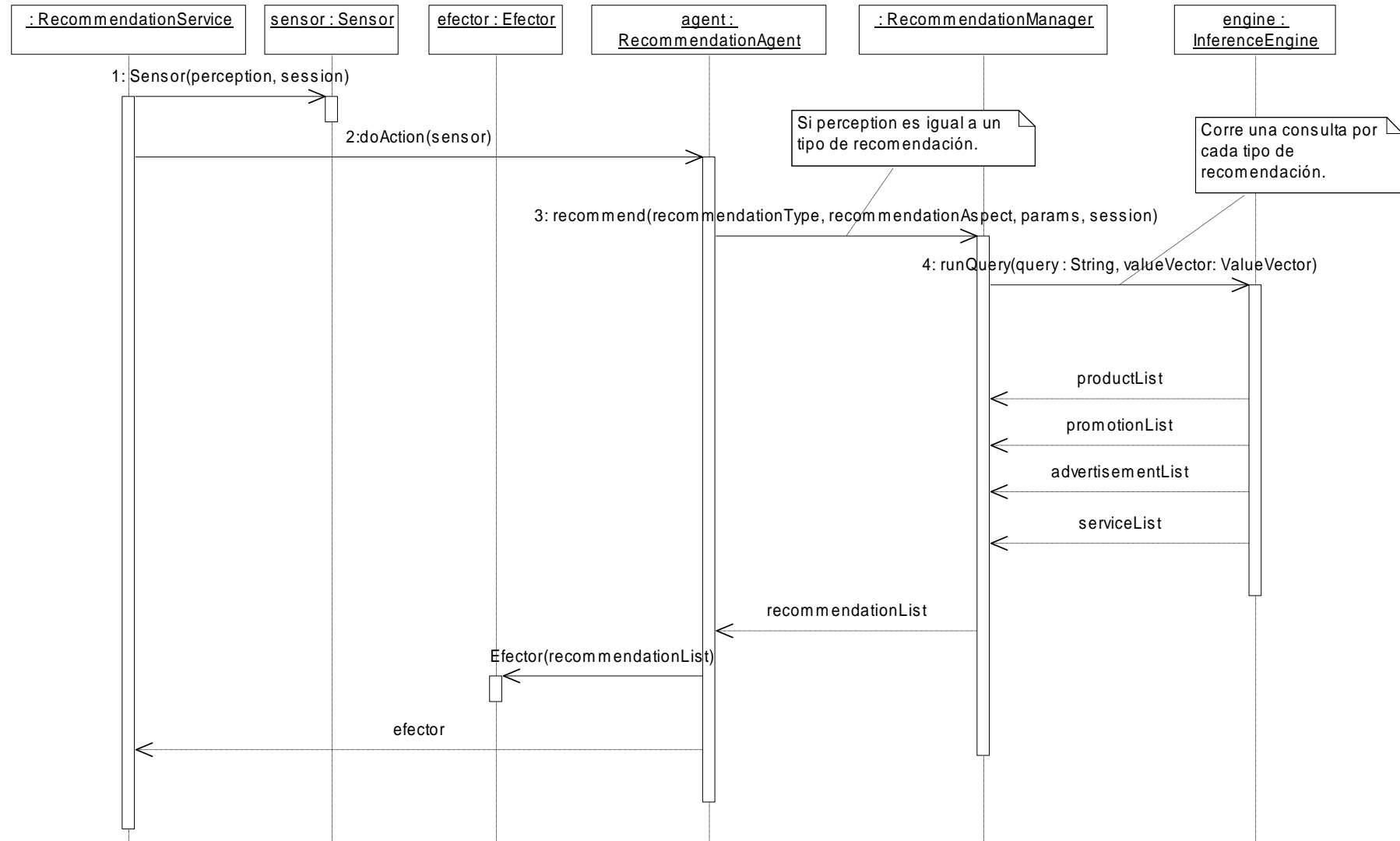


Figura 2.7 Diagrama de secuencia del Agente de Recomendación (Los Autores).

2.3.1.3.2. Modelo de la ontología del Agente

La mayoría de estereotipos de este modelo sirven para representar relaciones semánticas entre clases y se describen a continuación (Tabla 2.26):


Estereotipo	Descripción
<<ontology>>	Paquete UML que representa una ontología.
<<include>>	Relación de inclusión entre dos paquetes de ontologías.
<<is-a>> 	Relación de generalización (Hiponimia).
<<has-part>>	Relación de composición (Holonimia)
<<has-a>>	Relación de agregado con cardinalidad simple (Holonimia)
<<has-many>>	Relación de agregado con cardinalidad múltiple (Holonimia)

Tabla 2.26 Estereotipos para elementos de ontologías(Los Autores).

La ontología esta compuesta de varios paquetes de clases descritos en la Tabla 2.27 y en la Figura 2.8 se muestra su nivel de dependencia. Es necesario mencionar que el diseño de la ontología de aplicación depende del sistema con el cual va a interactuar por lo tanto su especificación no corresponde a la fase de diseño.

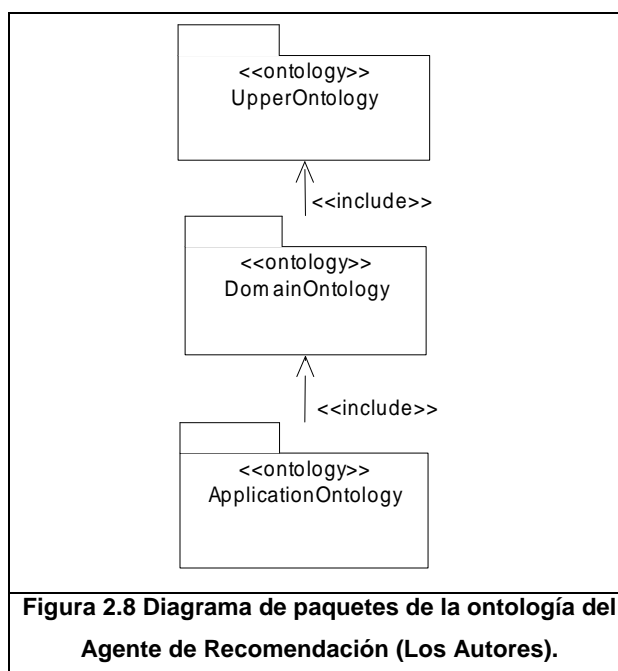


Figura 2.8 Diagrama de paquetes de la ontología del Agente de Recomendación (Los Autores).

Paquete	Descripción
UpperOntology	Ontología general.
DomainOntology	Ontología de dominio.
ApplicationOntology	Ontología de aplicación.

Tabla 2.27 Modelo de la ontología del Agente (Los Autores).

En las Tabla 2.28 y Tabla 2.29 se describen las clases definidas dentro de los diferentes paquetes de la ontología (Ver Figura 2.9) y en la Tabla 2.30 se presentan la descripción de las propiedades (*slots*) de cada clase.

Clase	Descripción	Rol
:THING	Clase con mayor nivel de generalización.	Abstracta
Entity	Representa una entidad abstracta o del mundo real.	Abstracta
Taxonomy	Representa una taxonomía.	Abstracta
Aspect	Generaliza dos aspectos de la aplicación referentes a entidades de dominio o de sistema.	Abstracta
TemporalEntity	Entidad que solamente reside en memoria durante el funcionamiento del Agente.	Abstracta
TemporalSystemEntity	Entidad temporal referente al sistema.	Abstracta
PersistentEntity	Entidad que reside en un medio almacenamiento físico.	Abstracta
System	Aspecto referente a la estructura del Agente.	Abstracta
Domain	Aspecto referente al dominio que cubre el Agente.	Abstracta
PersistentSystemEntity	Entidad persistente del sistema.	Abstracta
TemporalDomainEntity	Entidad temporal del dominio del agente.	Abstracta
DomainTaxonomy	Representa una taxonomía en el dominio de la aplicación.	Abstracta
PersistentDomainEntity	Entidad persistente del dominio de la aplicación.	Abstracta

Tabla 2.28 Clases de la ontología general: Paquete UpperOntology (Los Autores).

Clase	Descripción	Rol
CommerceEntity	Entidad que representa todas las entidades del dominio de e-commerce.	Abstracta
Category	Categoría de productos.	Concreta
Product	Producto.	Concreta
RecommendableEntity	Entidad recomendable. Generaliza producto, publicidad, servicio, promoción	Abstracta
ProductAssociatedEntity	Generaliza a publicidad, servicios y promociones asociadas a productos	Abstracta

Promotion	Promoción.	Concreta
Advertisement	Publicidad.	Concreta
Service	Servicio.	Concreta
Profile	Perfil de usuario.	Concreta
User	Usuario.	Concreta
Session	Sesión de usuario.	Concreta

Tabla 2.29 Clases de la ontología de dominio: Paquete DomainOntology (Los Autores).

Clase	Propiedad (Slot)	Descripción
PersistenEntity	code	Código o clave primaria.
	name	Nombre de la entidad.
	description	Descripción de la entidad.
PersistentDomainEntity	enabled	Bandera que indica si la entidad está habilitada o no.
DomainTaxonomy	parentCode	Código de la instancia padre.
Session	sessionId	Identificador de sesión de usuario.
	sessionTime	Fecha y hora en que se inicia la sesión de usuario.
User	password	Contraseña.

Tabla 2.30 Propiedades de clases (Los Autores).

Relación	Descripción
categoryHasProductAssociatedEntity	Una categoría tiene varias entidades asociadas a producto.
categoryHasProduct	Una categoría tiene varios productos.
productHasPart	Un producto tiene partes o accesorios.
productHasProductAssociatedEntity	Un producto tiene varias entidades asociadas a producto.
sessionHasUser	Una entidad de sesión de usuario tiene un usuario.
userHasProfile	Un usuario tiene perfil.
profileHasCommerceEntity	Un perfil tiene varias entidades de comercio.

Tabla 2.31 Relaciones entre clases (Los Autores).

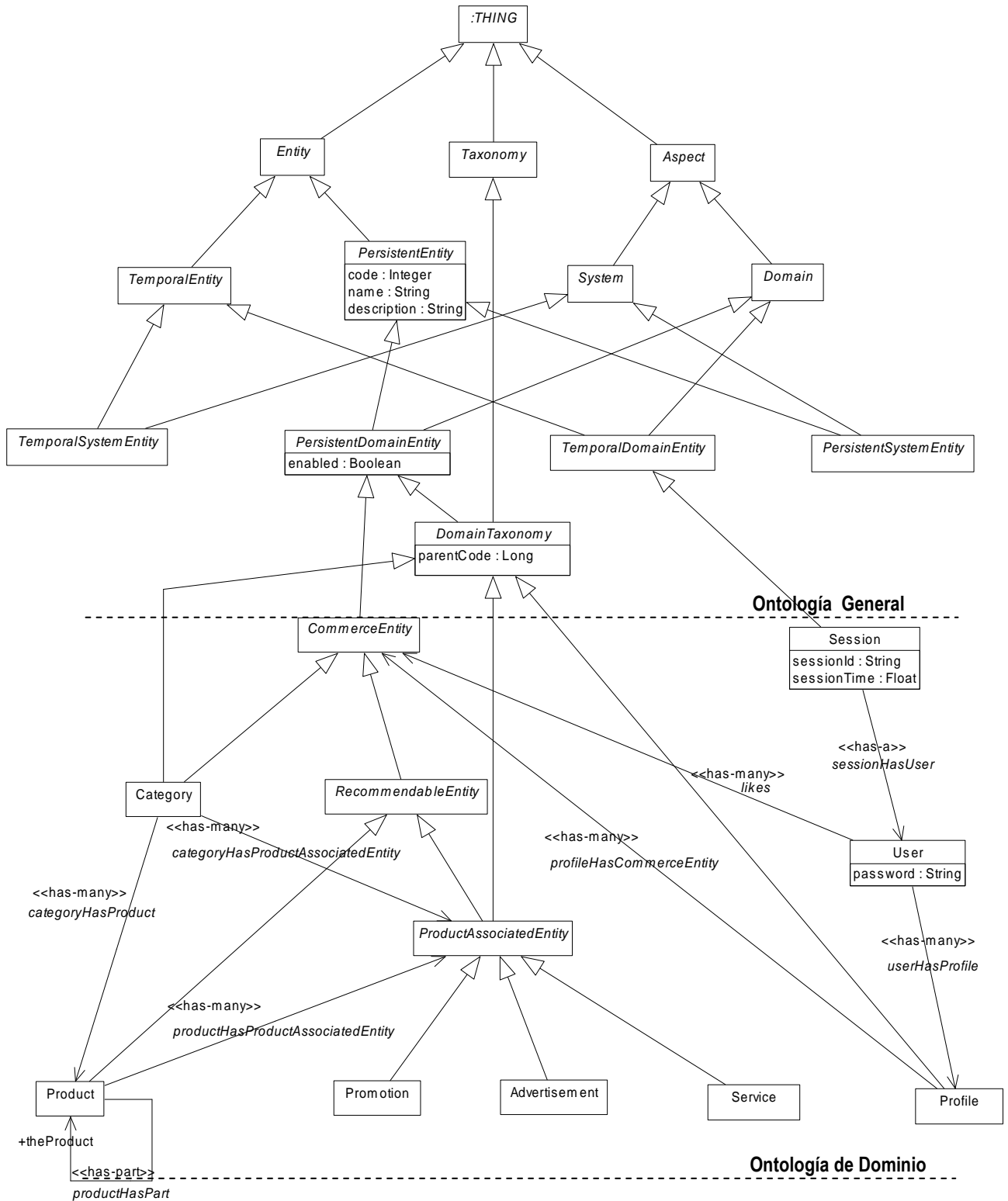


Figura 2.9 Ontología del Agente de Recomendación (Los Autores).

2.3.1.3.3. Modelo de reglas

Una ontología tiene elementos especiales como funciones y axiomas (§1.2.1.3) te y representan elementos del motor basado en reglas. Ver Anexo VIII.

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS DEL AGENTE

3.1. IMPLEMENTACIÓN

3.1.1. SELECCIÓN DE HERRAMIENTAS DE IMPLEMENTACIÓN

3.1.1.1. Plataformas de programación

3.1.1.1.1. J2EE

Edición empresarial del paquete Java creada y distribuida por Sun Microsystems. Comprende un conjunto de especificaciones y funcionalidades orientadas al desarrollo de aplicaciones empresariales.

Ventajas

- Java es un lenguaje multiplataforma y puede ejecutarse en cualquier sistema operativo que esté soportado por tecnología Java.
- Permite la fácil integración de APIs desarrollados sobre Java.
- Los más importantes proyectos relacionados con ontologías y motores de inferencia y están desarrollados sobre esta tecnología.

Desventajas

- No existen IDE que tenga una integración uniforme de todas las tecnologías desarrolladas.

3.1.1.1.2. Microsoft.NET

Plataforma de desarrollo de software de Microsoft creada para dar ciertas facilidades como énfasis en transparencia de redes, con independencia de plataforma y que permita un desarrollo rápido de aplicaciones.

Ventajas

- Posee la herramienta VisualStudio.NET que es un IDE que integra uniformemente todas las tecnologías relacionadas con este framework.

Desventajas

- No existe muchas opciones de tecnologías relacionadas con ontologías desarrolladas sobre esta plataforma.

3.1.1.2. Plataformas para ontologías

3.1.1.2.1. *OpenCyc*

Es la versión OpenSource del proyecto Cyc que incluye una ontología y motor de inferencia. La versión 1.0 tiene un banco de datos con 47000 conceptos con sus definiciones y relaciones.

Ventajas

- Posee una base de conocimiento con ontología general y ontologías de dominio ya desarrolladas.
- Navegador Web con la documentación de la base de conocimiento.
- Base de conocimiento y motor de inferencia integrados.
- Motor de inferencia permite razonamiento con encadenamiento hacia atrás y hacia adelante.

Desventajas

- Existe poca documentación y ejemplos acerca del funcionamiento del motor de inferencia este proyecto.
- Existen pocos proyectos desarrollados sobre esta tecnología.
- No tiene un IDE.

3.1.1.2.2. *Kaon-II*

Es la segunda versión del proyecto KAON de la Universidad de Karlsruhe el cual consiste en SDK para desarrollo de ontologías además tiene un motor de inferencia integrado.

Ventajas

- Tiene un componente que utiliza registros de cualquier base de datos como si fueran instancias de ontologías y las tablas las interpreta como clases.
- Motor de inferencia integrado.
- Soporte de formatos RDF y OWL.
- Motor de inferencia permite razonamiento con encadenamiento hacia atrás y hacia adelante.

Desventajas

- No tiene una ontología general desarrollada previamente.
- Existe poca documentación y ejemplos acerca del funcionamiento del

motor de inferencia de este proyecto.

- No existen proyectos desarrollados sobre esta tecnología.
- No tiene un IDE.

3.1.1.2.3. *Protégé*

Herramienta y API que permite la edición de bases de conocimiento y ontologías. Es un proyecto open-source y tiene una arquitectura extensible para la creación de herramientas personalizadas basadas en conocimiento.

Ventajas

- Posee un IDE avanzado para desarrollo de ontologías.
- Soporta diferentes formatos de almacenamiento de ontologías como por ejemplo XML, RDF, OWL entre otros.
- Es una plataforma extensible que permite es desarrollo de plugins.
- Existen plugins de integración con motores de inferencia como JESS, CLIPS y Algernon.
- Existe gran cantidad de documentación disponible.

Desventajas

- No tiene un motor de inferencia integrado.
- No posee una integración con bases de datos.

3.1.1.3. **Motores de inferencia**

3.1.1.3.1. *CLIPS*

Es una herramienta de desarrollo productivo para sistemas expertos que provee un ambiente completo para la construcción de sistemas basados en objetos o reglas. Fue creado en 1985, y actualmente es utilizado académica e industrialmente debido a sus características.

Ventajas

- Existen una gran cantidad de proyectos y ejemplos desarrollados.
- Está ampliamente documentado.

Desventajas

- El plugin de CLIPS para Protégé no funciona exclusivamente en Java sino que utiliza las librerías de CLIPS sobre la plataforma Windows.

- El motor de inferencia solamente tiene razonamiento con encadenamiento hacia adelante.

3.1.1.3.2. *JESS*

Es un motor de reglas y ambiente para Script desarrollado completamente sobre Java por en Sandia National Laboratories.

Ventajas

- Es inspirado en la tecnología CLIPS.
- Permite una total integración con tecnología Java.
- Tiene un IDE desarrollado sobre Eclipse.
- Es una tecnología desarrollada comercialmente lo que garantiza un gran nivel de escalabilidad y estabilidad.
- Existe bastante documentación disponible.

Desventajas

- El motor de inferencia solamente tiene razonamiento con encadenamiento hacia adelante.
- Para trabajar con la base de conocimiento de Protégé las instancias deben transformarse a hechos JESS.

3.1.1.3.3. *Algernon*

Es un sistema inferencia basado en reglas implementado en Java e integrado con Protégé.

Ventajas

- Permite una total integración con tecnología Java.
- En motor de inferencia trabaja directamente con la base de conocimiento de Protégé. No hace una transformación de la base de conocimiento.
- El motor de inferencia tiene razonamiento con encadenamiento hacia delante y hacia atrás.

Desventajas

- Existe poca documentación disponible.
- Se encuentra todavía en desarrollo y no tiene su funcionalidad completamente implementada.

3.1.1.4. Formato de almacenamiento de la base de conocimiento

3.1.1.4.1. *Formato nativo de Protégé*

Consiste en archivos de texto con sintaxis en lenguaje CLIPS. Es necesario aclarar que inicialmente Protégé fue idealizado como un editor gráfico de bases de conocimiento para CLIPS; sin embargo con la evolución de la plataforma se aumentan elementos adicionales que hacen que este formato ya no sea compatible con CLIPS actualmente.

Ventajas

- Su estructura facilita la creación intuitiva de ontologías gráficamente.
- Este formato es el más eficiente de la plataforma.
- Está soportado por JESS.

Desventajas

- Su estructura no soporta el ingreso de axiomas que también forma parte de las ontologías.

3.1.1.4.2. *OWL*

Ventajas

- Su estructura permite la especificación de axiomas dentro de la misma ontología.
- Está soportado por JESS.
- Cuenta con un validador de consistencia lógica para la ontología.
- Utiliza mecanismos de inferencia avanzados diseñados para OWL.

Desventajas

- En un formato que no es eficiente con la plataforma Protégé.
- Se necesitan conocimientos avanzados de lógica de descripción para la especificación de axiomas.
- El plugin de Protégé para OWL tiene errores y funcionalidad pendiente por implementar.

3.1.1.5. Entorno de desarrollo integrado (IDE)

3.1.1.5.1. *Eclipse*

Es un entorno independiente de la plataforma open-source para crear

aplicaciones cliente de cualquier tipo. La primera y más importante aplicación que ha sido realizada con este entorno es el IDE Java y compilador que se usaron para desarrollar el propio Eclipse.

Ventajas

- Permite la programación con diferentes lenguajes. El IDE de Java tiene funciones avanzada para programación de código.
- Tiene amplia variedad de plugins para diferentes tecnologías.
- Tiene un plugin que facilita el desarrollo de servicios Web con tecnología Java.

Desventajas

- Como está desarrollado sobre Java en ocasiones ocupa grandes recursos de procesador.
- Para el desarrollo de aplicaciones Web se necesita grandes recursos de memoria.

3.1.1.5.2. NetBeans

Es un IDE open-source de Sun Microsystems escrito enteramente en Java. Sun ONE Studio llamado inicialmente Forte for Java se basó en este proyecto y es la versión comercial.

Ventajas

- Tiene un buen soporte para desarrollo de aplicaciones Web.
- Tiene amplia variedad de plugins para diferentes tecnologías.

Desventajas

- Como está desarrollado sobre Java en ocasiones ocupa grandes recursos de procesador.
- Para el desarrollo de aplicaciones Web se necesita grandes recursos de memoria.

3.1.1.6. Servidor Web y servicios Web

Dado que J2EE provee de los estándares y especificaciones para la publicación de aplicaciones y servicios Web, la comparación de alternativas de tecnologías no es crítica ya que cualquier aplicación desarrollada sobre J2EE es capaz de funcionar con cualquier tecnología que cumpla con las especificaciones.

Las principales tecnologías para publicación de servicios Web en plataforma J2EE son:

- AXIS y el servidor Apache Tomcat
- Java Web Services Development Pack (JWSDP) de Sun Microsystems (basado en Jakarta Tomcat)
- Novell exteNd (basado en la plataforma J2EE)
- WebSphere

Tipo de herramienta	Selección/Justificación
Plataforma de programación	Se seleccionó la plataforma J2EE ya que los proyectos más importantes relacionados con ontologías están desarrollados sobre esta plataforma.
Plataforma para ontologías	Se seleccionó la plataforma Protégé debido a que posee un IDE avanzado para el desarrollo de ontologías y permite seleccionar el motor de inferencia que se va a utilizar.
Motor de inferencia	Se seleccionó la tecnología JESS debido a que es un motor de inferencia estable y además está bien documentado. Por otro lado, está implementado en Java, con lo que se conserva la uniformidad de la plataforma de desarrollo del Agente. Es necesario aclarar que en la fase de diseño se determinó la necesidad de un motor de inferencia con razonamiento híbrido por la razón de que se necesitan determinar relaciones de herencia en una taxonomía de categorías de productos y en una taxonomía de perfiles de usuario hecha en base a datos de mercado. Es necesario aclarar que JESS solamente implementa un algoritmo de razonamiento progresivo. Sin embargo, las tareas que necesitan razonamiento regresivo pueden ser emuladas en este motor de inferencia haciendo un preprocesado de las relaciones de herencia de las taxonomías en la ontología del Agente.
Formato de almacenamiento de la base de conocimiento	Se escoge el formato nativo de Protégé para el almacenamiento de la base de conocimiento ya es un formato simple, eficiente y estable que puede funcionar con el motor de inferencia JESS. Además para el desarrollo del Agente solamente se necesitan mecanismos básicos de inferencia.
Entorno de desarrollo integrado	La elección del IDE de programación no es tan crítica porque solamente sirve para la escritura de código Java; sin embargo se eligió Eclipse ya que existe un IDE de JESS desarrollado para esta plataforma y además tiene un plugin para desarrollo de servicios Web de una manera fácil.
Servidor Web y servicios Web	Se selecciona las plataformas AXIS para servicios Web y el servidor Web Tomcat de Apache Foundation ya que existe un plugin de Eclipse que presta facilidades para desarrollo de servicios Web con estas tecnologías.

Tabla 3.1 Selección de herramientas de implementación (Los Autores).

3.1.2. DESCRIPCIÓN DE HERRAMIENTAS Y PLATAFORMA DE DESARROLLO

El ARPS está desarrollado completamente sobre Java ya que permite una fácil integración de diferentes tecnologías afines. En la Tabla 3.2 se presenta la descripción del software utilizado mientras que la Figura 3.1 se muestra la plataforma de desarrollo del Agente.

Software	Descripción	Versión	Sitio Web
Eclipse	Herramienta de desarrollo	3.2	www.eclipse.org
J2EE	Plataforma de programación	1.4	java.sun.com
Protégé	Herramienta y API para ontologías	3.1.1	protege.stanford.edu
JESS	Consola y API para reglas	6.1p	herzberg.ca.sandia.gov/jess
JessTab	Plugin y API de integración Protégé-Jess	12	www.ida.liu.se/~her/JessTab/
Apache AXIS	Plataforma de Servicios Web	1.3	ws.apache.org/axis
Apache Tomcat	Servidor Web	5.0	jakarta.apache.org/tomcat

Tabla 3.2 Descripción de herramientas y plataforma de desarrollo (Los Autores).

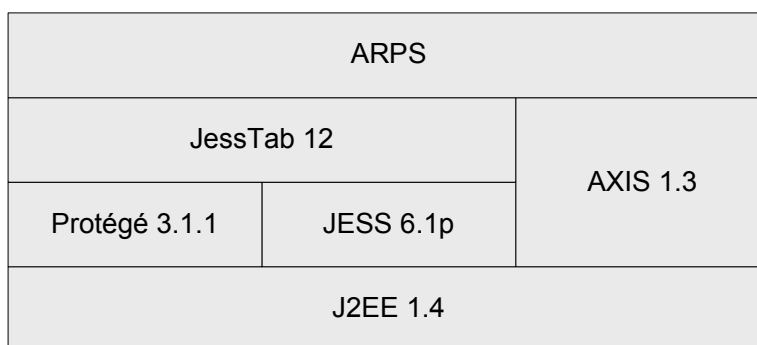


Figura 3.1 Plataforma de desarrollo del APRS (Los Autores)

3.1.2.1. Eclipse

Es una plataforma de código abierto para el desarrollo de aplicaciones cliente complejas. Sin embargo, ha sido extensamente utilizada para el desarrollo de entornos de desarrollo integrado IDE (*Integrated Development Environment*) como por ejemplo el ampliamente difundido JDT (*Java Development Kit*). También cuenta con el WTP (*Web Tools Project*) que es un conjunto de utilidades para

desarrollo de aplicaciones J2EE y tecnologías Web (Figura 3.2).

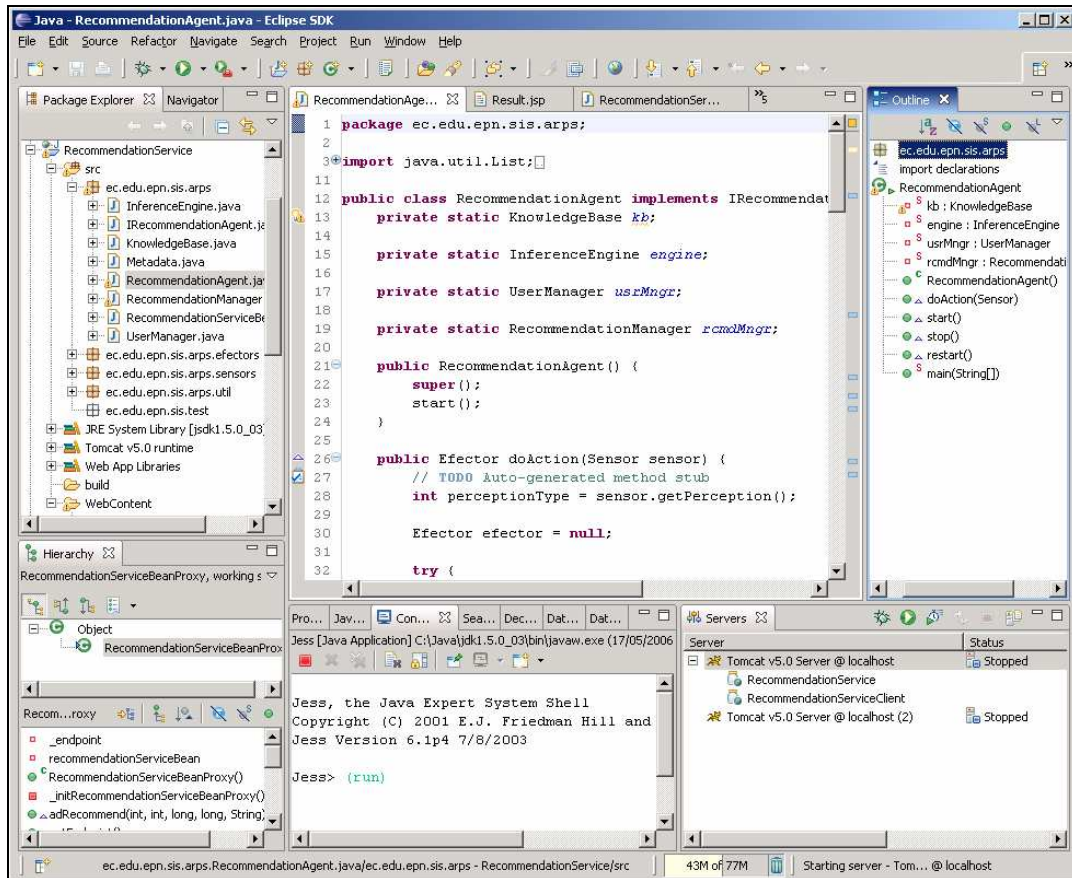


Figura 3.2 Eclipse IDE (Los Autores)

3.1.2.2. Java 2 Platform, Enterprise Edition (J2EE)

Es una plataforma de programación para desarrollo y ejecución de aplicaciones distribuidas de arquitectura multicapa. Está basada en componentes de software modulares que se ejecutan en servidores de aplicaciones e incluyen algunas especificaciones de API para conectividad de redes y bases de datos, componentes de negocio, sitios y servicios Web; lo que permite la creación de aplicaciones de negocio portables y escalables al mismo tiempo que permite la integración con otras tecnologías.

3.1.2.3. Protégé

Es una herramienta integrada de software (Figura 3.3) usada por desarrolladores y expertos de dominio para crear sistemas basados en conocimiento. Protégé cuenta con las siguientes características:

- Interfaz gráfica integrada y extensible.

- API de programación basado en tecnología Java.
- Soporte de diferentes formatos de almacenamiento (BDD ,XML, RDF, OWL).

Las aplicaciones desarrolladas con Protégé se usan en resolución de problemas y toma de decisiones en un dominio en particular. La interfaz de usuario permite las siguientes funcionalidades:

1. Modelado de una ontología de clases que describen un tema en particular.
2. Creación de herramientas de adquisición del conocimiento para ingreso de información.
3. Creación de instancias específicas de datos y de una base de conocimiento.
4. Ejecución de aplicaciones.

La herramienta de adquisición del conocimiento permite a los expertos de dominio el ingreso fácil y natural de su conocimiento en determinada área. La base de conocimiento resultante puede ser utilizada con un PSM (*Problem Solving Method*) para responder preguntas y resolver problemas respecto a un dominio. Por último, el producto final es una aplicación de software que utiliza la base de conocimiento para resolver problemas del usuario final empleando métodos de resolución de problemas, sistemas expertos o sistemas de soporte de decisiones.

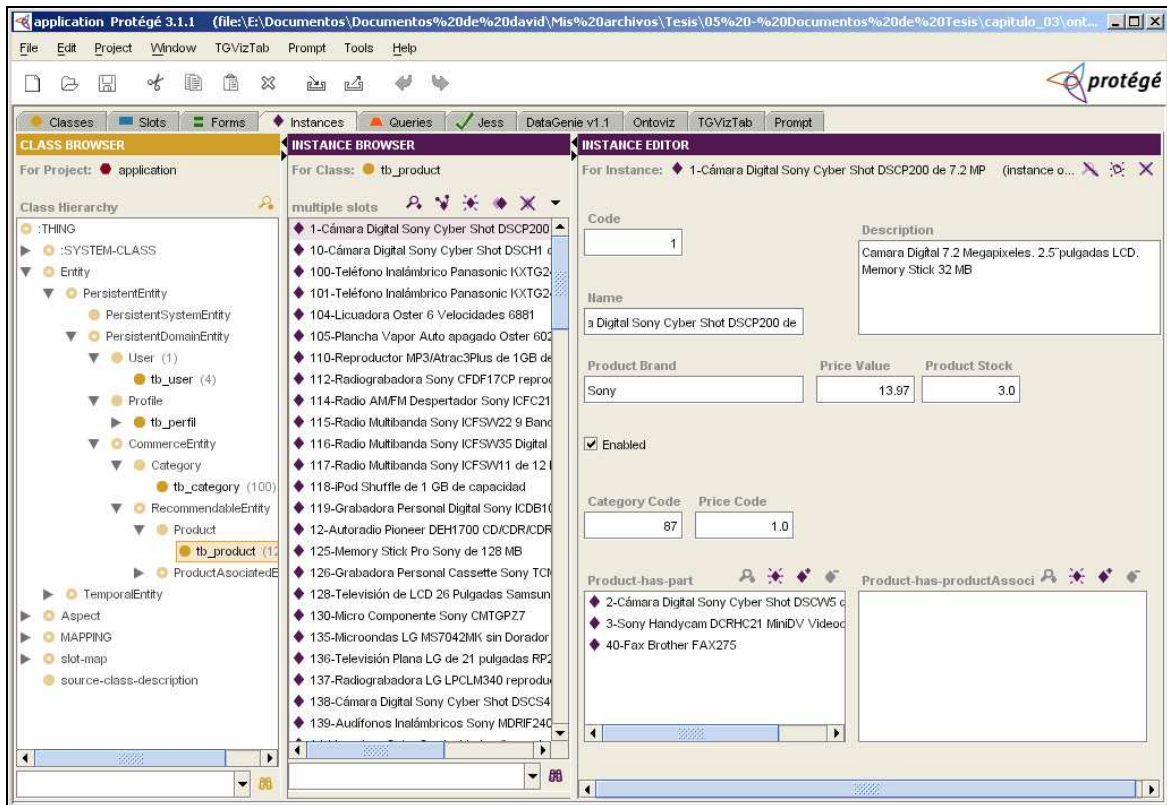


Figura 3.3 Interfaz gráfica de usuario de Protégé (Los Autores).

3.1.2.4. Java Expert Shell System (JESS)

Es un *shell* de sistemas expertos compuesto de un motor de reglas y un lenguaje inspirado mayor parte en CLIPS. JESS está desarrollado en Java de manera que tiene acceso a todas las facilidades de esta tecnología como por ejemplo comunicación de redes, gráficos y acceso a bases de datos.

El motor de reglas está compuesto de tres partes:

- Motor de inferencia,
- Base de reglas y
- Memoria de trabajo (base de hechos).

El motor de reglas de JESS utiliza una versión mejorada del conocido algoritmo Rete para enlazar reglas en la memoria de trabajo. El motor de inferencia está conformado por el enlazador de patrones, la agenda y el motor de ejecución. En el enlazador de patrones aplica las reglas a la memoria de trabajo para construir la agenda. El motor de ejecución dispara las reglas de la agenda la cual cambia el contenido de la memoria de trabajo y reinicia el ciclo (Figura 3.4).

Por otro lado, JESS ha sido utilizado para desarrollar una amplia gama de

software comercial como por ejemplo:

- Sistemas expertos que evalúan reclamos de seguros y aplicaciones a préstamos.
- Detectores de intrusos de red y auditores de seguridad.
- Software para sistemas de control.
- Servidores que ejecutan reglas del negocio.
- Sitios de e-commerce inteligentes.
- Juegos.

Estudios han demostrado que JESS es rápido y eficiente sin embargo su desempeño es sensitivo a la operación de recolección de memoria basura de Java por tanto en la mayoría de casos es necesario configurar la máquina virtual JVM (*Java Virtual Machine*).

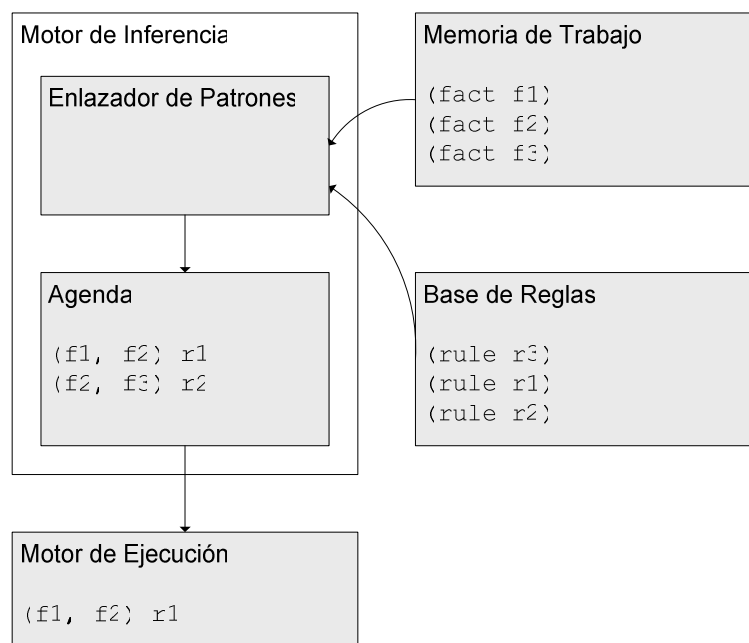


Figura 3.4 Arquitectura de JESS (Friedman-Hill 2003).

3.1.2.5. JessTab

Es un *plugin* y API desarrollado para Protégé que provee integración con JESS. Contiene una ventana con una consola JESS embebida dentro de un tab de Protégé y un conjunto de extensiones de JESS que permiten mapear las bases de conocimiento a la base de hechos de JESS y la manipulación de los mismos.

Esta herramienta permite el desarrollo de bases de conocimiento gráficamente y

la ejecución de métodos de resolución de problemas y de esta forma facilita el desarrollo de aplicaciones más complejas.

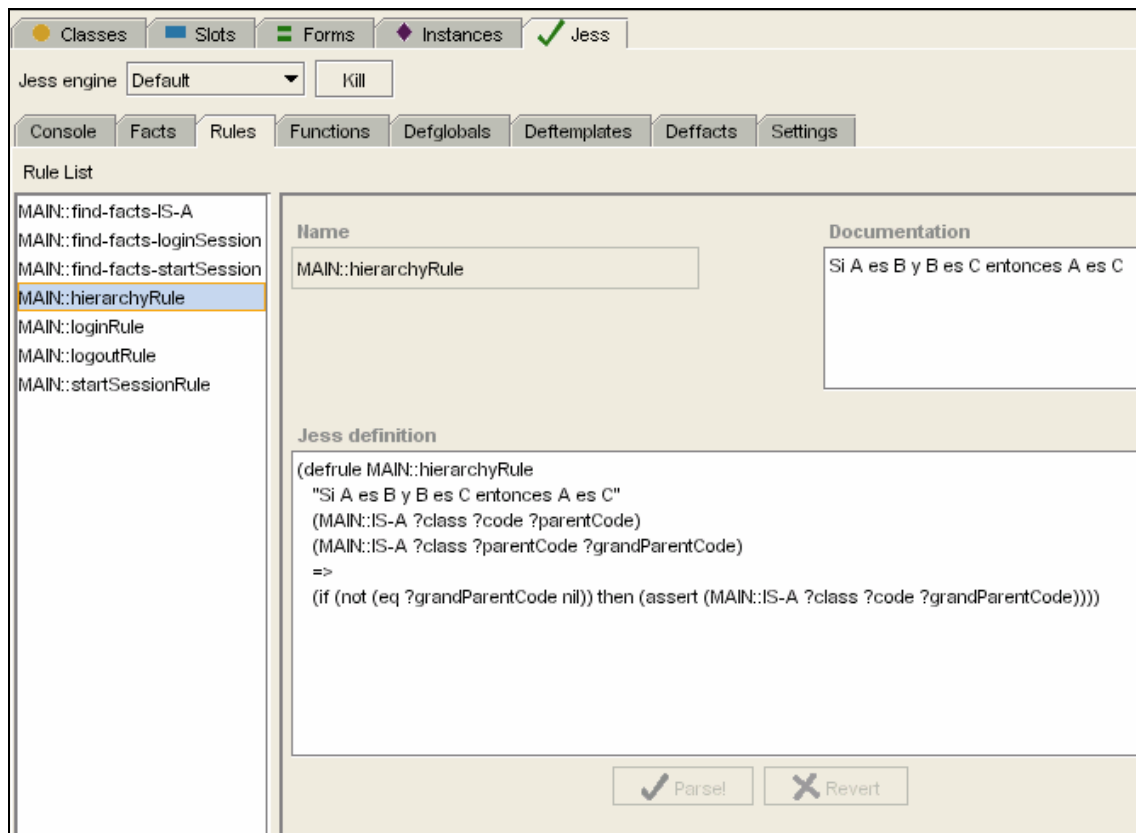


Figura 3.5 JessTab (Los Autores).

3.1.2.6. Apache Tomcat

Es un contenedor Web desarrollado por Apache Software Foundation. Implementa servlets y páginas JSP (*Java Server Pages*) especificadas en J2EE. De esta manera se tiene un ambiente par ejecutar código Java en un servidor Web. Está dotado de herramientas de administración y configuración; sin embargo puede ser configurado con la edición de los archivos de configuración que generalmente están en formato XML.

3.1.2.7. Servicios Web y Apache AXIS

Un servicio Web es una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes y ejecutadas sobre cualquier plataforma pueden utilizar los servicios Web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la

adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares. (Tomado de Wikipedia 2006).

3.1.2.7.1. *Estándares empleados*

- **Web Services Protocol Stack:** Así se denomina al conjunto de servicios y protocolos de los servicios Web.
- **XML:** Es el formato estándar para los datos que se vayan a intercambiar.
- **SOAP o XML-RPC:** Protocolos sobre los que se establece el intercambio.
- **Otros protocolos:** los datos en XML también pueden enviarse de una aplicación a otra mediante protocolos normales como HTTP, FTP, o SMTP.
- **WSDL:** Es el lenguaje de la interfaz pública para los servicios Web. Es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios Web.
- **UDDI:** Protocolo para publicar la información de los servicios Web. Permite a las aplicaciones comprobar qué servicios Web están disponibles.
- **WS-Security:** Protocolo de seguridad aceptado como estándar por OASIS. Garantiza la autenticación de los actores y la confidencialidad de los mensajes enviados.

3.1.2.7.2. *Ventajas de los servicios Web*

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
- Permiten que servicios y software de diferentes compañías ubicadas en

diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.

- Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar.

3.1.2.7.3. *Inconvenientes de los servicios Web*

- Para realizar transacciones no pueden compararse en su grado de desarrollo con los estándares abiertos de computación distribuida como CORBA.
- Su rendimiento es bajo si se compara con otros modelos de computación distribuida, tales como RMI, CORBA, o DCOM. Es uno de los inconvenientes derivados de adoptar un formato basado en texto. Dado que entre los objetivos de XML no se encuentra la eficiencia de procesamiento.
- Al apoyarse en HTTP, pueden esquivar medidas de seguridad basadas en firewall cuyas reglas tratan de bloquear o auditar la comunicación entre programas a ambos lados de la barrera.
- Existe poca información de servicios Web para algunos lenguajes de programación

3.1.2.7.4. *Razones para crear servicios Web*

La principal razón para usar servicios Web es que se basan en HTTP sobre TCP en el puerto 80. Dado que las organizaciones protegen sus redes mediante firewalls, que filtran y bloquean gran parte del tráfico de Internet, cierran casi todos los puertos TCP salvo el 80, que es, precisamente, el que usan los navegadores. Los servicios Web se levantan por este puerto, por esta razón no resultan bloqueados.

Otra razón es que antes de que existiera SOAP, no había buenas interfaces para acceder a las funcionalidades de otros ordenadores en red. Las que había eran algunas propuestas poco conocidas.

Una tercera razón por la que los servicios Web son muy prácticos es que pueden aportar gran independencia entre la aplicación que usa el servicio Web y el propio servicio. De esta forma, los cambios a lo largo del tiempo en uno no deben afectar

al otro. Esta flexibilidad será cada vez más importante, dado que la tendencia a construir grandes aplicaciones a partir de componentes distribuidos más pequeños es cada día más acusada.

3.1.2.7.5. *Apache AXIS*

Es una plataforma de Servicios Web basada en Java y XML que consiste en la implementación de un servidor SOAP (*Simple Object Access Protocol*), varias utilidades y APIs para generar y distribuir servicios Web. Está implementado en C++ sin embargo existe una implementación de Apache Software Foundation que se integra con el servidor Web Tomcat.

3.1.3. SELECCIÓN DE VERSIONES DE HERRAMIENTAS

3.1.3.1. Versión de Protégé

Versión	Características	Observaciones
2.1	Es la versión más ampliamente documentada. Sobre esta versión se han realizado la mayoría de plugins.	Tiene algunos errores en el IDE.
3.1.1	Ofrece una interfaz gráfica renovada y más funcional que la versión 2.1. Es la última versión más estable liberada de Protégé	Se escogió esta versión por estabilidad y la funcionalidad de la interfaz.

3.1.3.2. Versión de JESS

Versión	Características	Observaciones
6.1p	Es una versión liviana de JESS.	Seleccionó esta versión porque es liviana y funciona perfectamente con la versión 3.1.1 de Protégé.
7.0b	Soporta OWL. Tiene un IDE desarrollado para Eclipse. Soporta formato de reglas en XML.	Esta versión es más pesada que la anterior. Además el IDE para Eclipse es inestable.

3.1.3.3. Versión de JessTab

Versión	Características	Observaciones
12	Funciona con la versión 3.1.1	Se escogió esta versión porque es la que viene por defecto con la versión 3.1.1 de Protégé
14	Soporta OWL, reglas en XML y JESS 7.0	Es una versión beta.

3.1.3.4. Versión de J2EE

Versión	Características	Observaciones
1.5	Es la última versión de J2EE	Protégé 1.3.1 está desarrollada sobre la versión 1.5

3.1.3.5. Versión de Eclipse

Versión	Características	Observaciones
3.2	Es la última versión estable de Eclipse. Funciona con JSDK 1.5	Se escogió esta versión ya que tiene el plugin WTP que ofrece facilidades para desarrollo aplicaciones Web y tiene un asistente para la realización de servicios Web de forma fácil.

3.1.3.6. Versión de Apache Tomcat

Versión	Características	Observaciones
5.0	Es una versión estable pero no es la última.	Se escogió esta versión ya que Apache AXIS es compatible con Tomcat 5.0
5.5	Es la última versión estable de Tomcat	No soporta AXIS.

3.1.3.7. Versión de Apache AXIS

Versión	Características	Observaciones
1.3	Es la última versión estable de AXIS	Es la versión que viene por defecto con el plugin WTP de Eclipse.

3.1.4. MODELO DE IMPLEMENTACIÓN

3.1.4.1. Estructura de paquetes Java

Por convención los nombres de los paquetes se nombran en base al URL invertido de la institución a la que pertenece el proyecto. En la Figura 3.6 se puede apreciar la estructura de los paquetes Java mientras que en la Tabla 3.4 se muestra la descripción de los contenidos de cada paquete.

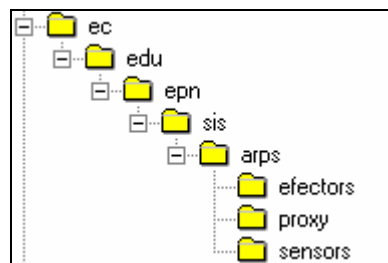


Figura 3.6 Estructura de paquetes Java.

Paquete Java	Paquete de diseño
ec.edu.epn.sis.arps	RecommendationAgent
ec.edu.epn.sis.arps.sensors	AgentSensor
ec.edu.epn.sis.arps.efectors	AgentEfector
ec.edu.epn.sis.arps.proxy	Contiene la clase Proxy que sirve para conectarse a un Servicio Web.
ec.edu.epn.sis.arps.util	Contiene utilidades para la creación de sentencias de código JESS.

Tabla 3.3 Descripción de paquetes Java (Los Autores).

Paquete	Archivo	Clase de diseño / Descripción
ec.edu.epn.sis.arps	IRecommendationAgent.java	IRecommendationAgent
	RecommendationServiceBean.java	RecommendationService
	RecommendationAgent.java	RecommendationAgent
	UserManager.java	UserManager
	RecommendationManager.java	RecommendationManager
	InferenceEngine.java	InferenceEngine
	KnowledgeBase.java	KnowledgeBase
Metadata.java	Metadata	
ec.edu.epn.sis.arps.sensors	Sensor.java	Sensor

	RecommendationSensor.java	RecommendationSensor
	UserSensor.java	UserSensor
ec.edu.epn.sis.arps.efectores	Efector.java	Efector
	RecommendationEfector.java	RecommendationEfector
ec.edu.epn.sis.arps.proxy	IRecommendationBean.java	Clases generadas automáticamente por Eclipse.
	RecommendationBeanService.java	
	RecommendationBeanProxy.java	
	RecommendationBeanSoapBindingStub.java	
ec.edu.epn.sis.arps.util	JessStatement.java	Clase que representa una sentencia JESS.

Tabla 3.4 Estructura de paquetes Java (Los Autores).

3.1.4.2. Estructura de la ontología

La ontología del ARPS está conformada por ontología general, de dominio y aplicación. Además va acompañada de varios archivos con código JESS con reglas acerca de los elementos de la ontología. En la Figura 3.7 se puede observar la estructura de los archivos Protégé mientras que en la Tabla 3.5 se muestra la descripción de los mismos.

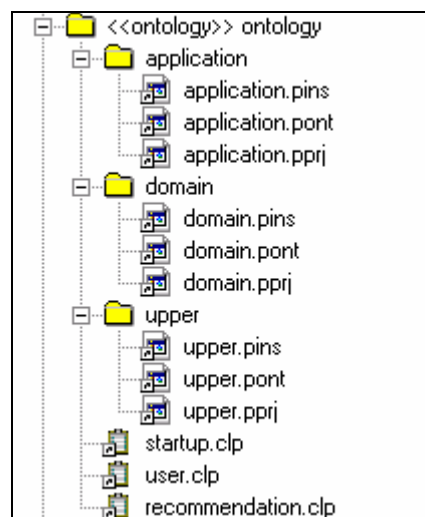


Figura 3.7 Estructura de la ontología.

Directorio	Archivo	Descripción
ontology	startup.clp	Código JESS de inicialización del motor de reglas.
	user.clp	Código con reglas de usuario.
	recommedation.clp	Código JESS con reglas de recomendación.
upper	upper.pins	Instancias de la ontología general.
	upper.pont	Ontología general.
	upper.pprj	Proyecto Protégé.

domain	domain.pins	Instancias de la ontología de dominio.
	domain.pont	Ontología de dominio.
	domain.pprj	Proyecto Protégé.
application	application.pins	Instancias de la ontología de aplicación.
	application.pont	Ontología de aplicación.
	application.pprj	Proyecto Protégé.

Tabla 3.5 Descripción de la ontología (Los Autores).

3.1.4.3. Estructura del directorio virtual del Servicio de Recomendación

Todo Servicio Web tiene un archivo WSDL (*Web Service Description Language*) en donde describe toda la funcionalidad del servicio de manera que otras aplicaciones puedan utilizarlo. Además, el Servicio de Recomendación puede ser accedido a través de su URL y mediante un navegador Web se puede observar su funcionalidad en formato XML. La dirección URL del Agente es la siguiente:

`http://<servidor>/RecommendationService/services/RecommendationServiceBean?method=<método>[&<parámetro>=<valor>...]`

y la dirección del archivo WSDL es la siguiente:

`http://<servidor>/RecommendationService/wsd/RecommenadtionServiceBean.wsdl`

La descripción de las etiquetas se muestran a continuación (Tabla 3.6 y Tabla 3.7):

Etiqueta	Descripción
<servidor>	Nombre del servidor en donde reside el Servicio de Recomendación
<método>	Nombre del método que se quiere llamar. Ver Tabla 3.7
<parámetro>	Nombre de parámetro del método. Pueden existir varios. Ver Tabla 3.8
<valor>	Valor del parámetro a ingresar.

Tabla 3.6 Etiquetas del URL del Servicio de Recomendación (Los Autores).

Método	Descripción
start	Iniciar el Servicio de Recomendación del Agente.
startSession	Iniciar una sesión anónima en el Agente.

login	Autenticar usuario en el Agente.
logout	Salir del Agente
productRecommend	Recomendar productos.
promotionRecommend	Recomendar promociones.
adRecommend	Recomendar publicidad.
serviceRecommend	Recomendar servicios.

Tabla 3.7 Métodos del Servicio de Recomendación (Los Autores).

Parámetro	Tipo de dato	Descripción
session	string	Identificador de sesión de usuario en la aplicación cliente.
username	string	Nombre de usuario
password	string	Contraseña
rcmdAspect	int	Aspecto de recomendación: 1 = Recomendación respecto a partes de productos 2 = Recomendación respecto a categorías 3 = Recomendación respecto a perfil de usuario
numberOfItems	int	Número máximo de ítems de recomendación que se desea obtener.
productCode	long	Código de producto. En caso de no ser utilizado en la recomendación su valor es cero.
categoryCode	long	Código de categoría. En caso de no ser utilizado en la recomendación su valor es cero.

Tabla 3.8 Parámetros de los métodos del Servicio de Recomendación (Los Autores).

Método	session	username	password	rcmdAspect	numberOfItems	productCode	categoryCode
start	—	—	—	—	—	—	—
startSession	Sí	—	—	—	—	—	—
login	Sí	Sí	Sí	—	—	—	—
logout	Sí	—	—	—	—	—	—
productRecommend	Sí	—	—	Sí	Sí	Sí	Sí

promotionRecommend	Sí	—	—	Sí	Sí	Sí	Sí
adRecommend	Sí	—	—	Sí	Sí	Sí	Sí
serviceRecommend	Sí	—	—	Sí	Sí	Sí	Sí

Tabla 3.9 Uso de parámetros del Servicio de Recomendación.

En la Figura 3.8 se puede observar la estructura del directorio virtual del Servicio de Recomendación mientras que en la Tabla 3.10 se muestra la descripción de sus elementos.



Figura 3.8 Estructura del directorio virtual del Servicio de Recomendación (Los Autores).

Directorio	Descripción
RecommendationService	Servicio de recomendación
wsdl	Directorio que contiene el archivo WSDL del servicio Web.
services	Directorio que agrupa etiquetas de servicios.
RecommendationServiceBean	Etiqueta del servicio de recomendación.

Tabla 3.10 Estructura del directorio virtual del Servicio de Recomendación (Los Autores).

3.1.4.4. Elementos de distribución

El ARPS se agrupa en dos archivos cuya estructura de muestra en la Figura 3.9 y su descripción se presenta en la Tabla 3.11. El archivo RecommendationService.war se lo instala en el servidor Web Tomcat mientras que el archivo arps-proxy.jar debe ser embebido en cualquier aplicación J2EE que necesite de los servicios de recomendación del Agente. Cabe resaltar que para la conexión con el servicio desde otra plataforma, por ejemplo Microsoft .NET, es necesario generar otra clase proxy en base al archivo RecommendationServiceBean.wsdl.

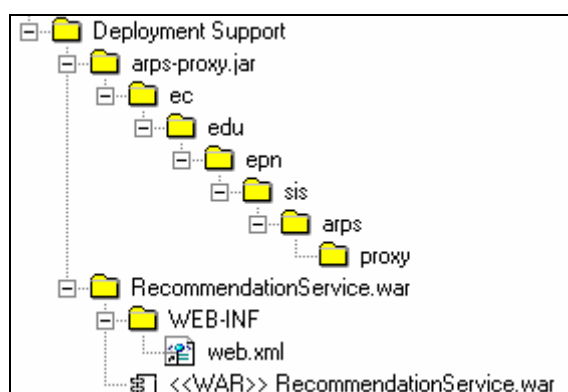


Figura 3.9 Estructura de los elementos de distribución (Los Autores).

Elemento	Descripción
arps-proxy.jar	Empaqueta la clase proxy que es necesaria para la conexión al Agente desde sistemas externos.
RecommendationService.war	Empaqueta todo el Servicio de Recomendación

Tabla 3.11 Descripción de los elementos de distribución (Los Autores).

En la Figura 3.10 se muestra el diagrama de distribución final del Agente.

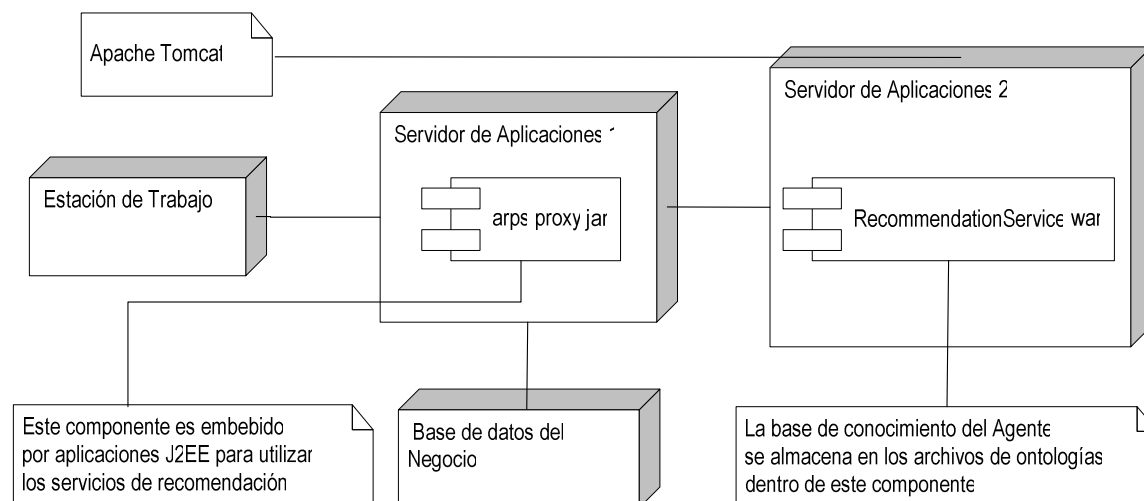


Figura 3.10 Diagrama de distribución final (Los Autores).

3.1.5. DETALLES DE IMPLEMENTACIÓN

3.1.5.1. Implementación de la ontología

De acuerdo a la metodología planteada en el capítulo uno (sección) se realizaron los siguientes pasos:

1. Determinar el dominio y alcance de la ontología
2. Considerar el uso de ontologías
3. Enumerar términos importantes
4. Definir la jerarquía de clases
5. Definir las propiedades de las clases
6. Definir las restricciones de las propiedades
8. Crear instancias

3.1.5.1.1. Determinar el dominio y alcance de la ontología

Con respecto al dominio, la ontología se define dentro del ámbito del e-commerce. En cuanto al alcance en concordancia con el agente de recomendación, la ontología representa los productos, servicios, promociones y publicidad de un

negocio electrónico y los perfiles que representan una taxonomía de la segmentación de mercado.

3.1.5.1.2. *Considerar el uso de ontologías*

La implementación de esta ontología fue realizada en su totalidad por los autores de este proyecto.

3.1.5.1.3. *Enumerar términos importantes*

En la Tabla 3.12 se generó un listado de las palabras o sustantivos más importantes relacionados al agente de recomendación:

Categoría de productos	Usuario	Producto
Cliente	Sesión	Servicio
Perfil de usuario	Publicidad	Promoción
Código	Nombre	Descripción
Recomendación	E-commerce	

Tabla 3.12 Conceptos principales del dominio de comercio electrónico (Los Autores)

3.1.5.1.4. *Definir la jerarquía de clases*

En base a las clases definidas en la Tabla 2.28, Tabla 2.29 y al diagrama de clases de ontología en la Figura 2.26, se procede a la creación de la ontología en el Protégé de la siguiente manera:

- Protege permite visualizar las diferentes opciones de configuración en la como CLASES, SLOTS, INSTANCIAS, FORMULARIOS a través de pestañas, cabe resaltar que plugins como JESS se presentan de igual manera.
- En la pestaña **Clases** se define y se estructura de manera gráfica las clases con sus respectivos **slots**, como se muestra en la Figura 3.11.

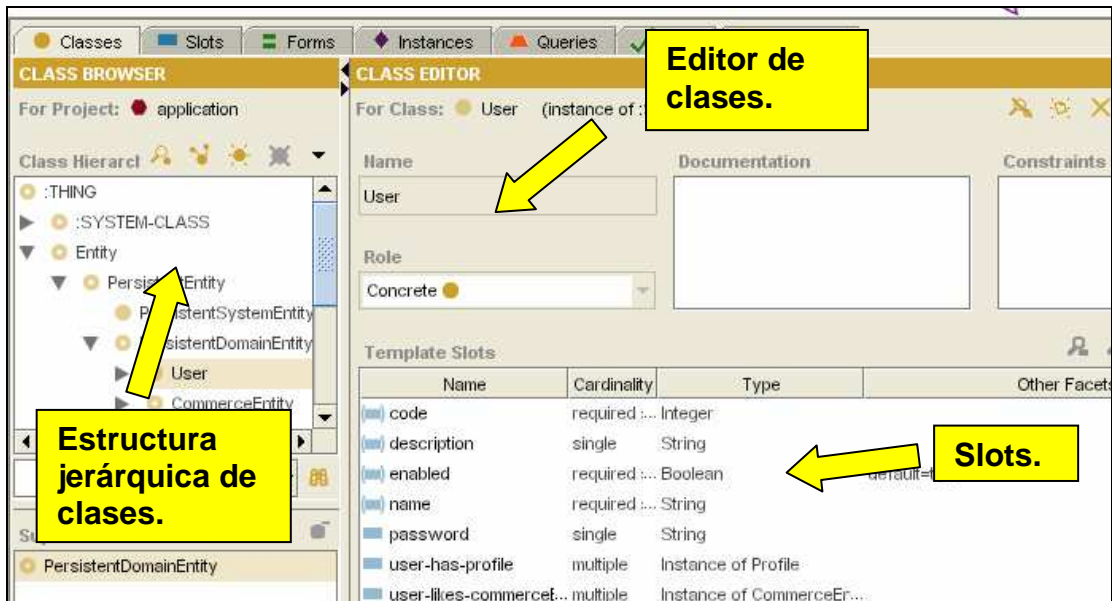


Figura 3.11 Editor de Clases (Los Autores)

En la Figura 3.12 se muestra la pantalla de configuración de una clase.

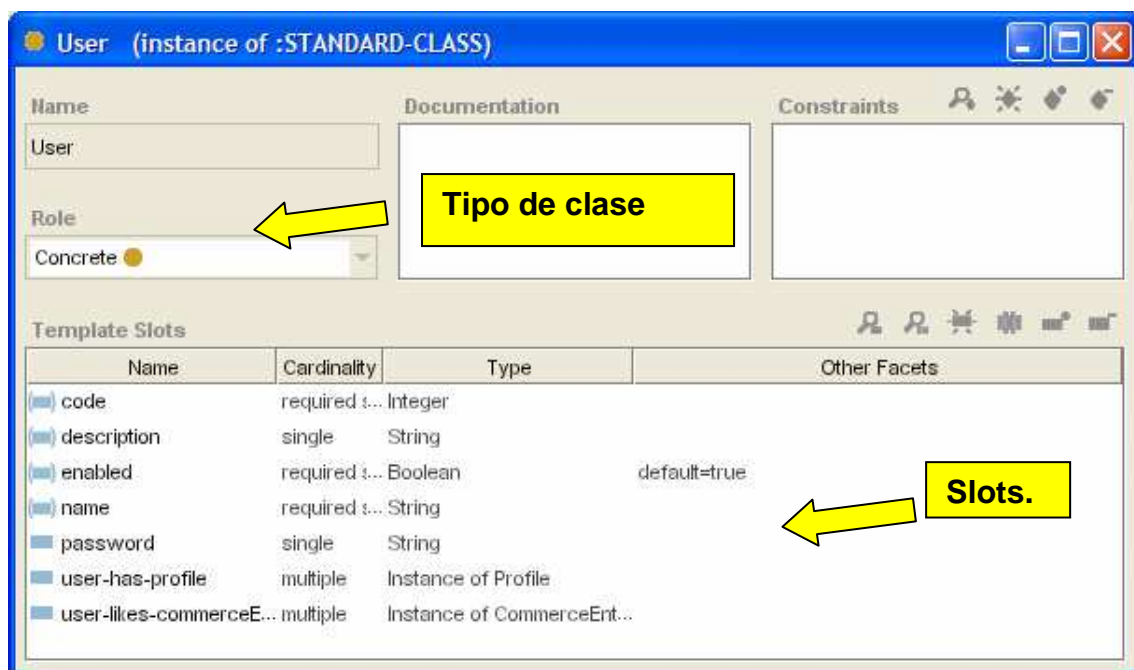


Figura 3.12 Pantalla de configuración de una Clase(Los Autores).

3.1.5.1.5. Definir las propiedades de las clases

Una vez definidas las clases se procede a definir las propiedades o slots de cada clase, estos proporcionan el comportamiento de la clase e implementan las relaciones entre clases, a continuación se muestra en la Figura 3.13.

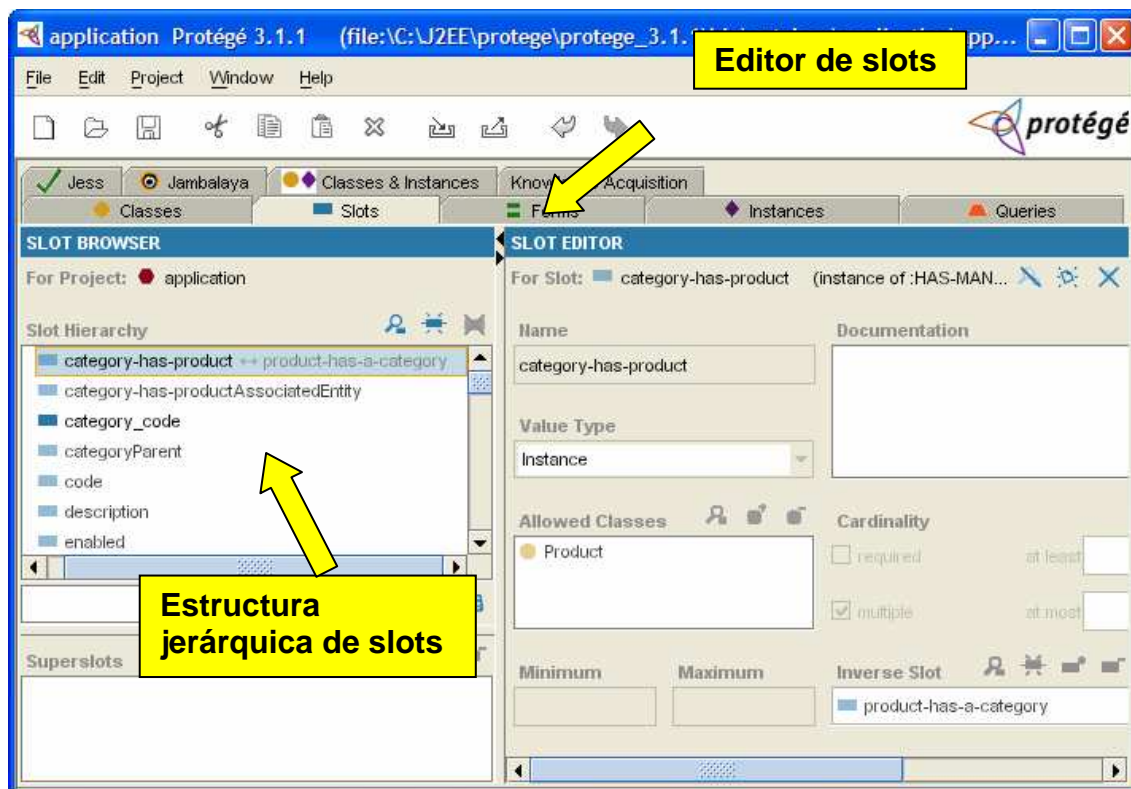


Figura 3.13 Editor de Slots(Los Autores).

3.1.5.1.6. Definir las restricciones de las propiedades

De acuerdo al tipo de slot se definen las restricciones, cardinalidad, valores por defecto, mínimos y máximos; en la Figura 3.14 se muestran las opciones:

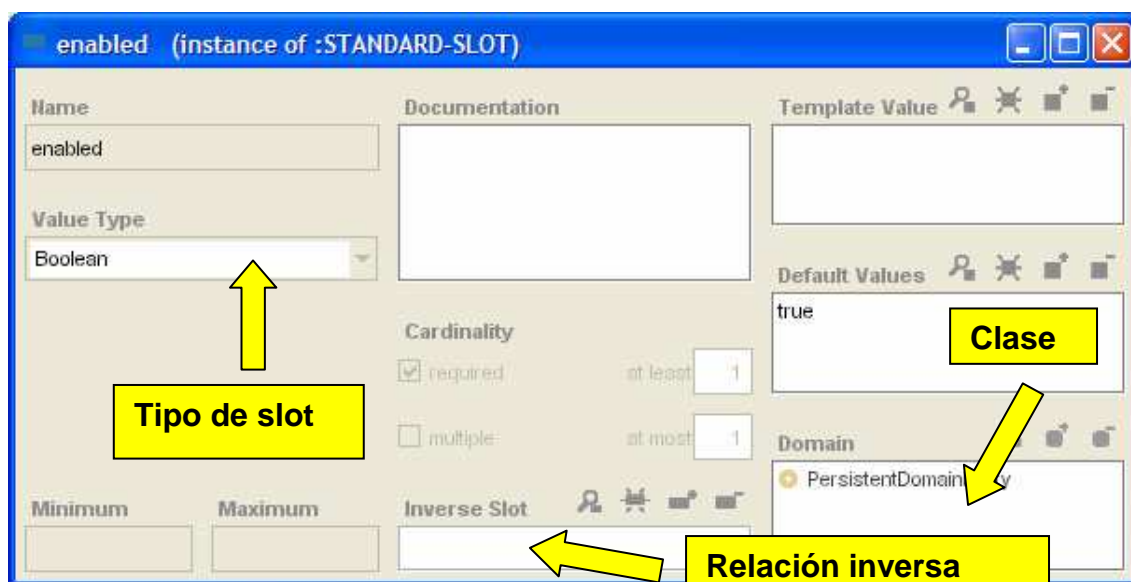


Figura 3.14 Pantalla de Configuración de Slots.

3.1.5.1.7. Crear instancias

A partir de las clases y propiedades configuradas, Protege genera formularios que

permiten ingresar manualmente instancias de las clases.

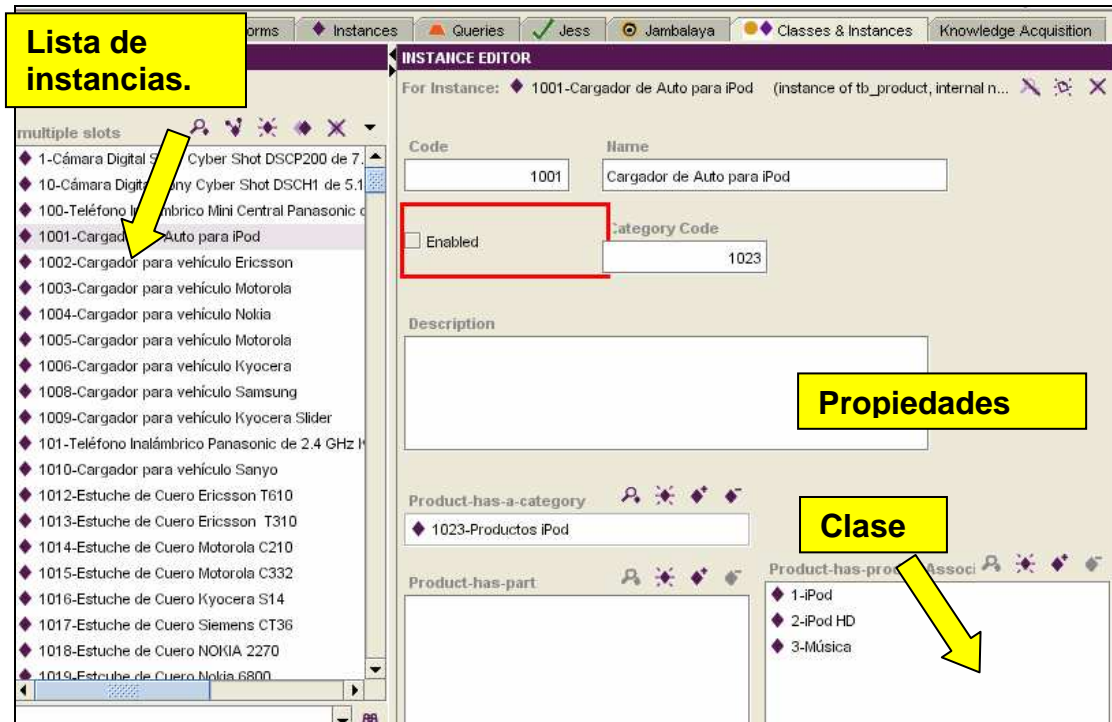


Figura 3.15 Editor de instancias (Los Autores)

3.1.5.2. Integración de Protégé y JESS

Para la integración del motor de reglas JESS con la plataforma de ontologías Protégé es necesario descargar el archivo JessTab12.jar que es API de integración entre Protégé y JESS de la página herzberg.ca.sandia.gov/jess. Este archivo debe estar en el mismo directorio de los archivos protege.jar y jess.jar.

Para cargar la ontología de aplicación dentro del motor de reglas JESS se creó el archivo de script startup.clp. Con el siguiente comando se carga la ontología de aplicación:

```
(load-project "./ontology/application/application.pprj")
```

Con el comando MAPCLASS se cargan las clases e instancias de la ontología como hechos JESS. Para cargar todas las clases e instancias se especifica a la clase de mayor jerarquía en este caso la clase :THING:

```
(mapclass :THING)
```


Una vez cargadas las clases y las instancias ya es posible trabajar con la ontología desde JESS. Las funciones de JessTab que se utilizan son las siguientes:

Función	Descripción
class	Obtiene el nombre de la clase de una instancia.
do-for-all-instances	Ejecuta una acción para todas las instancias que satisfacen una consulta.
do-for-instance	Ejecuta una acción para la primera instancia que satisface una consulta.
load-project	Carga un proyecto de Protégé
make-instance	Crea una instancia Protégé desde Jess
mapclass	Mapea una clase Protégé al sistema Jess.
slot-get	Obtiene el valor de un slot de una instancia de la base de conocimiento de Protégé
slot-set	Ingresa el valor de un slot de una instancia en la base de conocimiento de Protégé
subclassp	Determina si una clase Protégé es subclase de otra.
unmake-instance	Borra las instancias especificadas.

Tabla 3.13 Funciones de JessTab (Los Autores)

3.1.5.3. Implementación del código JESS

Los algoritmos más complejos de la funcionalidad del Agente se implementan en JESS. En la Tabla 3.14 se describen las funciones y elementos del lenguaje JESS que se utilizan para la implementación.

Función	Descripción
\$?args	Parámetro que significa que una función tiene un número de parámetros variable.
and	Retorna verdadero si ninguno de los argumentos retorna falso, caso contrario retorna falso.
assert	Afirma todos los hechos en la memoria de trabajo. Retorna un identificador del hecho.
batch	Ejecuta un archivo con código Jess

bind	Asigna un valor dado a una variable.
call	Llama un método Java de un objeto.
count-query-results	Corre una consulta cuyo nombre se da como parámetro y retorna el conteo de todas las coincidencias.
crif	Retorno de carro
deffunction	Definir una función
defquery	Definir una consulta
defrule	Definir una regla
div	División de enteros
eq	Igualdad
false	Falso
foreach	Bucle para iterar listas
if... then ... else	Condicional
implode\$	Crea una cadena de una lista
insert\$	Inserta uno o mas valores en una lista
length\$	Retorna el número de elementos en una lista
member\$	Retorna la posición de un valor en una lista, caso contrario retorna falso.
mod	Módulo
nil	Nulo
n o t	Negación
nth\$	Retorna el valor de un campo especificado en un lista de valores
printout	Imprime mensajes en consola
random	Generar número aleatorio
return	Retorna un valor para una función
str-cat	Concatena cadenas
time	Retorna el número de segundos desde el primero de enero de 1970
true	Verdadero
union\$	Unión de conjuntos
while... do...	Bucle de control

Tabla 3.14 Funciones y elementos del lenguaje JESS (Los Autores)

3.1.5.3.1. Archivo *startup.clp*

Con el objetivo de facilitar la programación en JESS se crearon algunas funciones de utilidad.

La función `code` obtiene el código de una instancia que pertenece a la clase `PersistentEntity` o sus subclases. Si la clase de la instancia `?entity` es subclase de la clase `PersistentEntity` entonces retornar el slot `code` de la instancia `?entity` caso contrario retornar nulo.

```
(deffunction code (?entity)
  (if (subclassp (class ?entity) PersistentEntity) then
    (return (slot-get ?entity code))
    else (return nil)))
```

La función `parentCode` obtiene el código del padre de una instancia que pertenece a la clase `Taxonomy` o sus subclases. Si la clase de la instancia `?entity` es subclase de `Taxonomy` entonces retornar el slot `parentCode` de la instancia `?entity` caso contrario retornar nulo.

```
(deffunction parentCode (?entity)
  (if (subclassp (class ?entity) Taxonomy) then
    (return (slot-get ?entity parentCode))
    else (return nil)))
```

La regla `hierarchyRule` es prácticamente el núcleo de inferencia del Agente debido a que representa al [silogismo de transitividad] "Si A es B y B es C entonces A es C". Para todas las instancias de las subclases de la clase `Taxonomy` se interpreta como "Si la categoría A pertenece a la categoría B y si la categoría B pertenece a la categoría C entonces la categoría A pertenece a la categoría C". La plantilla `IS-A` contiene los hechos en donde se especifica: la clase de una instancia, el código de la instancia y el código del padre de la instancia.

```
(defrule hierarchyRule "Si A es B y B es C entonces A es C"
  (IS-A ?class ?code ?parentCode)
  (IS-A ?class ?parentCode ?grandParentCode)
  =>
  (if (not (eq ?grandParentCode nil)) then
    (assert (IS-A ?class ?code ?grandParentCode))))
```

Con el siguiente script se crean hechos con la plantilla `IS-A` que sirven como base para las inferencias con la regla `hierarchyRule`. Para todas las instancias `?taxonomy` de clase `DomainTaxonomy` cuyo slot `parentCode` no es nulo, crear un

hecho de tipo IS-A.

```
(do-for-all-instances ((?taxonomy DomainTaxonomy))
  (not (eq (parentCode ?taxonomy) nil))
  (assert (IS-A (class ?taxonomy) (code ?taxonomy) (parentCode
?taxonomy))))
```

La función `isSubinstancep` devuelve verdadero o falso dependiendo si una instancia representa una subcategoría de otra.

Si la clase de la instancia `?subclass` es diferente de la clase de la instancia `?superclass` entonces retornar `false`.

Asignar a la variable `?hierarchyFact` el número de hechos IS-A de clase `?subclass` que tienen los códigos de `?subclass` y `?superclass`.

Si la variable `?hierarchyFact` es igual a 1 retornar verdadero, caso contrario retornar falso:

```
(deffunction isSubinstancep(?subclass ?superclass)
  (if (not (eq (class ?subclass) (class ?superclass))) then
    return FALSE)
  (bind ?hierarchyFact (count-query-results find-facts-IS-A (class
?subclass) (code ?subclass) (code ?superclass)))
  (if (eq ?hierarchyFact 1) then
    (return TRUE)
  else
    (return FALSE)))
```

La consulta `find-facts-IS-A` es usado por la función `isSubinstancep` y sirve para encontrar un determinado hecho de tipo IS-A.

```
(defquery find-facts-IS-A
  (declare (variables ?class ?subclass ?superclass))
  (IS-A ?class ?subclass ?superclass))
```

Con el comando `batch` se ejecutan los scripts con elementos referentes a usuarios y recomendaciones.

```
(batch user.clp)

(batch recommendation.clp)
```

3.1.5.3.2. Archivo *user.clp*

3.1.5.3.2.1. Iniciar sesión anónima

Si se afirma un hecho con plantilla `startSessionAssert` con código de sesión

?sessionId entonces crear una instancia de clase Session con código de sesión ?sessionId, y hora de sesión igual a la hora actual. Se imprime un mensaje que dice “El usuario a iniciado una sesión anónima” y a continuación el código de sesión.

```
(defrule startSessionRule "Iniciar sesión anónima"
  (startSessionAssert ?sessionId)
  =>
  (make-instance of Session (sessionId ?sessionId)
   (sessionTime (time)))
  (printout t "User has started anonymous session: " ?sessionId crlf))
```

3.1.5.3.2.2. Iniciar sesión de usuario

Si se afirma un hecho de plantilla loginRule entonces para todas las instancias de clase Session y de clase User en donde coinciden el código de sesión de usuario, el nombre de usuario y la contraseña, apuntar el slot ?session session-has-user a la instancia del usuario que coinciden nombre de usuario y password. Se imprimir un mensaje.

```
(defrule loginRule "Iniciar sesión de usuario"
  (loginAssert ?sessionId ?username ?password)
  =>
  (do-for-instance
   ((?session Session) (?user User))
   (and (eq ?username (slot-get ?user name))
        (eq ?password (slot-get ?user password))
        (eq ?sessionId (slot-get ?session sessionId)))
   (slot-set ?session session-has-user ?user)
   (printout t "Login " (slot-get ?session sessionId) " " (slot-get
?user name) crlf))
```

3.1.5.3.2.3. Terminar sesión de usuario

Si se afirma un hecho de plantilla logoutAssert asignar el hecho a la variable ?logoutAssert, entonces eliminar todas las instancias relacionadas con el código de sesión sessionId. Para la eliminación de algunos hechos relacionados con la sesión de usuario se necesitan las consultas find-facts-startSession y find-facts-loginSession.

```
;; Consulta de hechos startSession
(defquery find-facts-startSession
  (declare (variables ?sessionId))
  (startSessionAssert ?sessionId))

;; Consulta de hechos loginSession
(defquery find-facts-loginSession
  (declare (variables ?sessionId))
  (loginAssert ?sessionId ?u ?p))

;; Terminar sesión de usuario
```

```
(defrule logoutRule "Terminar sesión de usuario"
  ?logoutAssert <- (logoutAssert ?sessionId)
  =>
  (do-for-instance
    ((?session Session))
    (eq ?sessionId (slot-get ?session sessionId))
    (bind ?it (run-query find-facts-startSession ?sessionId))
    (while (?it hasNext)
      (bind ?token (call ?it next))
      (bind ?fact (call ?token fact 1))
      (retract ?fact))
    (bind ?it (run-query find-facts-loginSession ?sessionId))
    (while (?it hasNext)
      (bind ?token (call ?it next))
      (bind ?fact (call ?token fact 1))
      (retract ?fact))
    (retract ?logoutAssert)
    (printout t "Logout " (slot-get (slot-get ?session session-has-
user) name) crlf)
    (unmake-instance ?session)))
```

3.1.5.3.3. Archivo *recomendacion.clp*

Predicado que indica si una cosa pertenece a una taxonomía de instancias.

Devuelve verdadero si la instancia ?thing está en el slot ?taxonomy-has-thing de la instancia ?subtaxonomy y la instancia ?subtaxonomy es subinstancia de ?taxonomy. Caso contrario devuelve falso.

```
(deffunction thingOwnsToTaxonomyP (?thing ?taxonomy ?subtaxonomy
?taxonomy-has-thing)
  (and (member$ ?thing (slot-get ?subtaxonomy ?taxonomy-has-thing))
    (or (eq ?subtaxonomy ?taxonomy)
      (isSubinstancep ?subtaxonomy ?taxonomy))))
```

La función getCodeList, dado un instance-set de n columnas (?step) devuelve una lista con los códigos de las entidades que se quiere consultar (Primera columna).

Se usa en todas las funciones de recomendación.

```
(deffunction getCodeList (?instance-set ?step)
  (bind ?length (div (length$ ?instance-set) ?step))
  (bind ?codeList (create$))
  (bind ?i 0)
  (while (< ?i ?length) do
    (bind ?code (code (nth$ (+ 1 (* ?step ?i)) ?instance-set)))
    ;;(printout t ?code ' ')
    ;;(printout t (code (nth$ (+ 2 (* ?step ?i)) ?instance-set))
crlf)
    (bind ?codeList (insert$ ?codeList 1 ?code))
    (bind ?i (+ ?i 1)))
  (return ?codeList))
```

La función `getColumnList`, dado un `instance-set` de `n` columnas (`?step`), devuelve una lista con la columna de las entidades que se quiere consultar (`?column`). Se usa en todas las funciones de recomendación.

```
(deffunction getColumnList (?instance-set ?step ?column)
  (bind ?length (div (length$ ?instance-set) ?step))
  (bind ?codeList (create$))
  (bind ?i 0)
  (while (< ?i ?length) do
    (bind ?code (code (nth$ (+ ?column (* ?step ?i)) ?instance-
set)))
    ;;(printout t ?code ' ')
    ;;(printout t (code (nth$ (+ 2 (* ?step ?i)) ?instance-set))
crlf)
    (bind ?codeList (insert$ ?codeList 1 ?code))
    (bind ?i (+ ?i 1)))
  (return ?codeList))
```

La función `getRandomEntities` sirve para obtener `n` entidades randómicamente y hace lo siguiente:

1. Crea una cadena con el nombre de la función y los argumentos que se quiere ejecutar.
2. Ejecutar el código de la cadena
3. Crear una lista de `instance-set` de recomendaciones en base a una función y sus parámetros
4. Crear una lista de números aleatorios `randomList`
5. Calcular el largo de la anterior lista
6. Si el largo es menor que el número de items solicitados (`numberOfItems`) entonces
 - El máximo número de aleatorios a calcular (`maxItems`) es el largo
 - caso contrario es (`numberOfItems`)
7. Mientras el largo de `randomList` sea menor que el máximo número de aleatorios
 - Generar aleatorio entre 0 y el largo de la lista de recomendaciones-1.
 - Sumar 1 al aleatorio para tener índices basados en 1.
8. Para cada elemento de `randomList`
 - Recuperar el código de la entidad recomendada
 - Y adjuntar el código en las lista de recomendaciones randómicas (`randomEntityList`).

```

(deffunction getRandomEntities (?numberOfItems ?function $?args)
  (bind ?fnList (create$ ?function))
  (bind ?fnList (insert$ ?fnList 2 $?args))
  (bind ?fnString (str-cat "(" (implode$ ?fnList) ")"))

  ;; Imprimir cadena de función a ejecutar
  (printout t "Ejecutar: " ?fnString crlf)

  (bind ?itemList (eval ?fnString))

  (bind ?length (length$ ?itemList))

  (if (< ?length ?numberOfItems) then
    (bind ?maxItems ?length)
  else
    (bind ?maxItems ?numberOfItems))

  (bind ?randomList (create$))
  (while (< (length$ ?randomList) ?maxItems) do
    (bind ?rand (mod (random) ?length))
    (bind ?rand (+ ?rand 1))
    (bind ?randomList (union$ ?randomList (create$ ?rand))))

  (bind ?randomEntityList (create$))
  (foreach ?item ?randomList
    (bind ?code (nth$ ?item ?itemList))
    (bind ?randomEntityList (union$ ?randomEntityList (create$
?code))))
  (return ?randomEntityList))

```

3.1.5.3.3.1. Recomendación de productos

Función de recomendación de partes y accesorios:

Asignar a la variable ?instance-set el conjunto de partes y accesorios del producto indicado:

```

(deffunction partProductRecommendationFn (?productId)
  (bind ?instance-set (find-all-instances
    ((?part Product) (?product Product))
    (and (member$ ?part (slot-get ?product product-has-part))
      (eq ?productId (code ?product)))))
  (getCodeList ?instance-set 2))

```


Función de recomendación de productos asociados a categoría:

Seleccionar todos los productos que:

1. El productId = código de producto
2. El código de producto es diferente al código del producto asociado a categoría
3. El producto el miembro de una categoría
4. El producto asociado a categoría es de la categoría o
5. El producto asociado es de una subcategoría.

```
(deffunction categoryProductRecommendationFn (?productId)
  (bind ?category-instance-set (find-instance
    ((?category Category) (?product Product))
    (and (eq ?productId (code ?product))
      (member$ ?product (slot-get ?category category-has-product))))))
  (bind ?category (nth$ 1 ?category-instance-set))
  ;;(printout t 'CategoryCode' (code ?category) (name ?category)
?category-instance-set crlf)
  (bind ?instance-set (find-all-instances
    ((?categoryAssociatedProduct Product) (?subcategory
Category))
    (and (not (eq ?productId (code ?categoryAssociatedProduct)))
      (thingOwnsToTaxonomyP ?categoryAssociatedProduct ?category
?subcategory category-has-product))))
  ;;(printout t 'Largo: ' (length$ ?instance-set) crlf)
  (getCodeList ?instance-set 2))
```

Función de recomendación de productos de interés

Seleccionar todos los productos que:

Están directamente incluidos en el slot profile-has-commerceEntity

o productos que pertenecen a una categoría que está en el slot profile-has-commerceEntity

o productos que pertenecen a una subcategoría de una categoría que está en el slot profile-has-commerceEntity

profile-has-commerceEntity es del perfil o un superperfil.

```

(defun interestedProductRecommendationFn (?sessionId)
  ;;Perfiles de usuario
  (bind ?profile-instance-set (find-all-instances
    ((?session Session) (?user User) (?profile Profile))
    (and (eq ?sessionId (slot-get ?session sessionId))
      (eq ?user (slot-get ?session session-has-user))
      (member$ ?profile (slot-get ?user user-has-profile))))))

  (bind ?profileList (getColumnList ?profile-instance-set 3 3))

  ;;Superperfiles
  (bind ?superprofile-instance-set (find-all-instances
    ((?profile Profile) (?superprofile Profile))
    (and (member$ (code ?profile) ?profileList)
      (isSubinstancep ?profile ?superprofile))))

  (bind ?profileList (union$ ?profileList (getColumnList ?superprofile-instance-set 2 2)))

  ;;Productos del perfil
  (bind ?product-instance-set (find-all-instances
    ((?profile Profile) (?product Product))
    (and (member$ (code ?profile) ?profileList)
      (member$ ?product (slot-get ?profile profile-has-commerceEntity))))))

  (bind ?productList (getColumnList ?product-instance-set 2 2))

  ;;Productos de las categorías del perfil
  (bind ?category-instance-set (find-all-instances
    ((?profile Profile) (?category Category))
    (and (member$ (code ?profile) ?profileList)
      (member$ ?category (slot-get ?profile profile-has-commerceEntity))))))

  (bind ?categoryList (getColumnList ?category-instance-set 2 2))
  ;;(printout t "CategoryList: " (length$ ?categoryList))

  (bind ?subcategory-instance-set (find-all-instances
    ((?category Category) (?subcategory Category))
    (and (member$ (code ?category) ?categoryList)
      (isSubinstancep ?subcategory ?category))))

  (bind ?categoryList (union$ ?categoryList (getColumnList ?subcategory-instance-set 2 2)))

  ;;(printout t "SubCategoryList: " (length$ ?categoryList))

  (bind ?categoryproduct-instance-set (find-all-instances
    ((?category Category) (?product Product))
    (and (member$ (code ?category) ?categoryList)
      (member$ ?product (slot-get ?category category-has-product))))))

  ;;(printout t "productList: " (length$ ?categoryproduct-instance-set))
  (bind ?productList (union$ ?productList (getColumnList ?categoryproduct-instance-set 2 2)))
)

```

3.1.5.3.3.2. Recomendación de entidades asociadas a partes o accesorios de producto.

Función de recomendación de entidades asociadas al producto o a sus partes y accesorios.

Seleccionar todas las entidades asociadas a producto que:

- Cuya clase sea igual a la del argumento
- El producto es el del argumento
- La entidad pertenece al producto o
- La entidad pertenece a un accesorio del producto

```
(deffunction partProductAssociatedEntityRecommendationFn (?productId
?class)
  (bind ?product-instance-set (find-instance
    ((?product Product))
    (eq ?productId (code ?product))))

  (bind ?productObject (nth$ 1 ?product-instance-set))

  (bind ?partList (slot-get ?productObject product-has-part))

  (bind ?partList (union$ ?partList ?product-instance-set))

  (bind ?instance-set (find-all-instances
    ((?part Product) (?productAssociatedEntity
ProductAssociatedEntity))
    (and (member$ ?part ?partList)
      (member$ ?productAssociatedEntity (slot-get ?part product-
has-productAssociatedEntity))
      (or (eq ?class (class ?productAssociatedEntity))
        (subclassp (class ?productAssociatedEntity) ?class))))))
  (getColumnList ?instance-set 2 2))
```

Función de recomendación de entidades asociadas a categoría

Seleccionar todas las entidades asociadas a categoría que:

- Cuya clase sea igual a la del argumento
- La categoría es la del argumento
- La entidad pertenece a la categoría o
- La entidad pertenece a una subcategoría.

```
(deffunction categoryAssociatedEntityRecommendationFn (?categoryId
?class)
  (bind ?category-instance-set (find-instance
    ((?category Category))
    (eq ?categoryId (code ?category))))

  (bind ?category (nth$ 1 ?category-instance-set))

  ;;(printout t "Code: " (code ?category) crlf)

  (bind ?instance-set (find-all-instances
    ((?productAssociatedEntity ProductAssociatedEntity)
(?subcategory Category))
    (and (or (subclassp (class ?productAssociatedEntity) ?class)
      (eq (class ?productAssociatedEntity) ?class))
      (thingOwnsToTaxonomyP ?productAssociatedEntity ?category
?subcategory category-has-productAssociatedEntity))))
  ;;(printout t "Largo: " (length$ ?instance-set))
  (getColumnList ?instance-set 2 1))
```

Función de recomendación de entidades de comercio de interés

Seleccionar todas las entidades de comercio que:

Están directamente incluidos en el slot `profile-has-commerceEntity` o entidades de comercio que pertenecen a una categoría que está en el slot `profile-has-commerceEntity` o entidades de comercio que pertenecen a una subcategoría de una categoría que está en el slot `profile-has-commerceEntity` `profile-has-commerceEntity` es del perfil o un superperfil.

```
(deffunction interestedProductAssociatedEntityRecommendationFn
(?sessionId ?class)
;;Perfiles de usuario
(bind ?profile-instance-set (find-all-instances
((?session Session) (?user User) (?profile Profile))
(and (eq ?sessionId (slot-get ?session sessionId))
(eq ?user (slot-get ?session session-has-user))
(member$ ?profile (slot-get ?user user-has-profile))))))

(bind ?profileList (getColumnList ?profile-instance-set 3 3))

;;Superperfiles
(bind ?superprofile-instance-set (find-all-instances
((?profile Profile) (?superprofile Profile))
(and (member$ (code ?profile) ?profileList)
(isSubinstancep ?profile ?superprofile))))

(bind ?profileList (union$ ?profileList (getColumnList
?superprofile-instance-set 2 2)))

;;Entidades del perfil
(bind ?instance-set (find-all-instances
((?profile Profile) (?productAssociatedEntity
ProductAssociatedEntity))
(and (member$ (code ?profile) ?profileList)
(or (subclassp (class ?productAssociatedEntity) ?class)
(eq (class ?productAssociatedEntity) ?class))
(member$ ?productAssociatedEntity (slot-get ?profile
profile-has-commerceEntity))))))

;;(printout t "Largo: " (length$ ?instance-set) crlf)
(getColumnList ?instance-set 2 2))
```

3.1.5.4. Creación del API del Agente en Eclipse

La funcionalidad básica del API Agente de Recomendación es transportar los parámetros de entrada a JESS, ejecutar los comandos del motor de reglas y recoger los resultados obtenidos para ser manejados con Java.

Eclipse tiene un IDE con facilidades para la programación en Java. En donde se realizaron los siguientes pasos:

1. Crear un nuevo proyecto Java llamado ARPS.

2. Añadir las librerías protege.jar, jess.jar y JessTab.jar que son parte de la plataforma de programación.
3. Crear el paquete ec.edu.epn.sis.arps y sus respectivos subpaquetes conforme al modelo de implementación.
4. Crear las clases de cada paquete conforme al modelo de implementación.
5. Crear propiedades y métodos para cada clase conforme al modelo de diseño.
6. Implementar los algoritmos de cada método conforme a los diagramas de secuencia del modelo de diseño.

La clase principal es RecommendationServiceBean.java, la cual tiene las propiedades y métodos que se van a utilizar con el servicio Web. Contiene un objeto de clase RecommendationAgent el cual recibe información a través objetos de clase Sensor y devuelve resultados a través de objetos de clase Efector.

En la clase Metadata.java solamente se almacenan constantes y parámetros que se utilizan en la programación.

Las clases RecommendationManeger.java y UserManager.java se comunican con la clase InferenceEngine solamente para ingresar parámetros y extraer resultados a través de sensores y efectores. Es necesario aclarar que en Java no se implementan algoritmos de mayor complejidad.

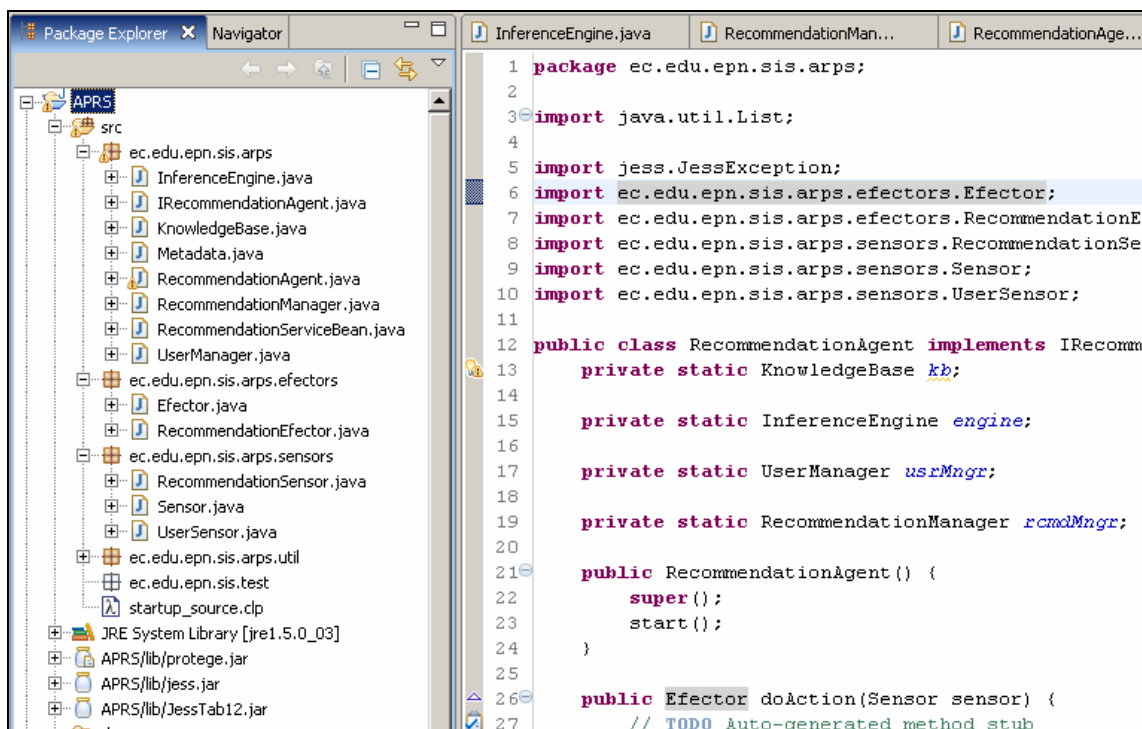


Figura 3.16 Estructura del API del Agente de Recomendación en Eclipse (Los Autores)

3.1.5.5. Integración JESS y Java

3.1.5.5.1. Archivo *InferenceEngine.java*

Rete es la clase principal del API de JESS, a través de ella se controla el motor basado en reglas. La clase *InferenceEngine* es el motor de inferencia del Agente y hereda todas las propiedades y métodos de la clase *Rete*. El método *start()* inicia el motor basado en reglas cargando las funciones de integración entre Protégé y JESS a través de la función *addUserpackage(new JessTabFunctions())*. Con la función *loadKnowledgeBase(kbFile)* se carga el archivo de Protégé con la ontología. Esta función crea programáticamente una cadena con la sentencia JESS

```
(load-project "./ontology/application/application.pprj")
```

Y carga los archivos de script de JESS con el comando

```
(batch startup.clp)
```

La clase *JessStatement* solamente tiene facilidades para la creación de cadenas con sintaxis JESS y pertenece al paquete *ec.edu.epn.sis.arps.util*

La función *executeCommand* pertenece a la clase *Rete* y ejecuta comandos JESS creados a partir de cadenas de texto. La clase *InferenceEngine* tiene una sobrecarga de esta función que en vez de utilizar cadenas de texto como argumento utiliza objetos de tipo *JessStatement*. La clase *Value* pertenece al API de JESS y representa cualquier tipo de dato que se maneja en este motor de reglas. En este objeto se recogen los resultados de cualquier función JESS y se los puede extraer para manejarlos a través de Java.

```
package ec.edu.epn.sis.arps;
import jess.JessException;
import jess.Rete;
import jess.Value;
import JessTab.JessTabFunctions;
import ec.edu.epn.sis.arps.util.JessStatement;

public class InferenceEngine extends Rete {
    private static final long serialVersionUID = -7085336575458678210L;

    public InferenceEngine(String kbFile) {
        super();
        start(kbFile);
    }

    public void start(String kbFile) {
        // 1. Añade funciones de integración de protege en JESS.
        addUserpackage(new JessTabFunctions());
        // 2. Carga base de conocimiento
        loadKnowledgeBase(kbFile);
    }
}
```

```

public void loadKnowledgeBase(String kbFile) {
    try {
        //1. (load-project "./ontology/application/application.pprj")
        JessStatement jessStmt = new JessStatement();
        jessStmt.append(Metadata.JESS_LOAD_PROJECT);
        jessStmt.spcAppend("\"");
        jessStmt.append(kbFile.trim());
        jessStmt.append("\"");
        jessStmt.putParenthesis();

        executeCommand(jessStmt);

        //2. (batch startup.clp)
        jessStmt = new JessStatement();
        jessStmt.append(Metadata.JESS_BATCH);
        jessStmt.spcAppend(Metadata.JESS_STARTUP_FILE);
        jessStmt.putParenthesis();

        executeCommand(jessStmt);
    } catch (JessException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

public Value executeCommand(JessStatement cmd) throws JessException {
    return super.executeCommand(cmd.toString());
}
}

```

3.1.5.6. Creación del servicio de Web

Ver Anexo VII.

3.2. CASO DE ESTUDIO

En el Anexo V se enuncia un caso de estudio y se presenta el proceso de adaptación del Agente de Recomendación. En base a este caso de estudio se obtienen los datos que servirán posteriormente para la realización de las pruebas.

3.3. PRUEBAS

Para la realización de las pruebas del Agente de Recomendación se utiliza la herramienta Web Services Explorer del Eclipse (Figura 3.17). En su panel izquierdo muestra una lista de todas las funciones del Servicio Web y al hacer clic se crean automáticamente formularios de ingreso de parámetros para estas funciones y en el panel inferior se muestran los resultados.

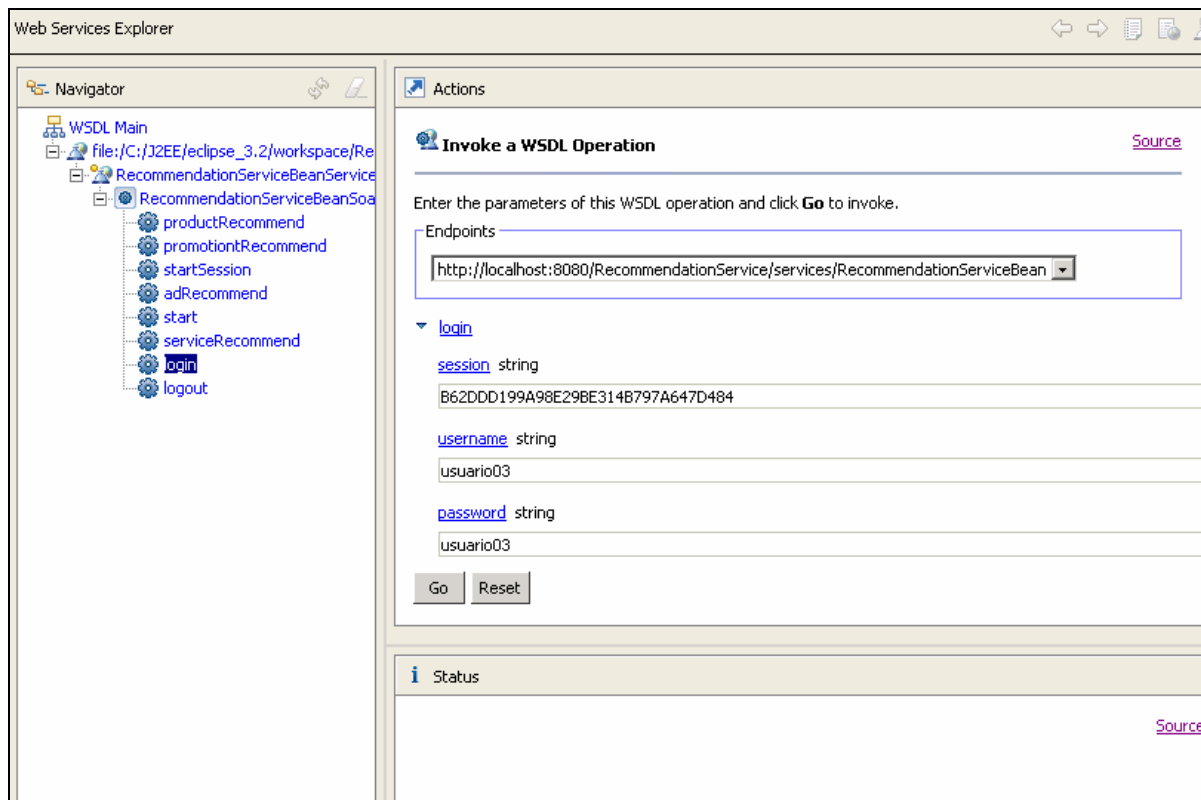


Figura 3.17 Web Services Explorer de Eclipse (Los Autores).

Por otra parte la plataforma de hardware y software utilizada para la realización de las pruebas se describe a continuación (Tabla 3.15):

Parámetro	Valor
Procesador	AMD Athlon
Velocidad	1333 Mhz
RAM	512 Mb
Sistema operativo	Windows XP Professional SP2

Tabla 3.15 Plataforma de hardware y software para realización de pruebas (Los Autores).

3.3.1. MODELO DE PRUEBAS PARA EL AGENTE DE RECOMENDACIÓN

3.3.1.1. Casos y procedimientos de pruebas

En el Anexo VI se presentan todos los casos y procedimientos de pruebas para el Agente de Recomendación.

3.3.1.2. Resultados de las pruebas

En la Tabla 3.16 se muestran los resultados de todos los casos de prueba del Agente de recomendación. Es necesario mencionar que las pruebas fracasaron en los siguientes casos:

- Recomendación de promociones, publicidad o servicios asociados a producto, partes o accesorios.
- Recomendación de promociones, publicidad o servicios asociados a la categoría.
- Recomendación de promociones, publicidad o servicios de interés.

Es los tres casos no se obtuvieron resultados y se observó que el porcentaje de uso del procesador sube al 100%.

Por este motivo se procedió a la revisión del código fuente del Agente. Se realizó una depuración paso a paso del código Java sin embargo no se encontraron errores.

Seguidamente, se procedió a la revisión del código JESS en donde no se encontró ningún error conceptual en la programación pero al realizar pruebas individuales en la consola se determinó que las funciones de recomendación en cuestión generaban explosiones combinatorias de instancias al realizar cálculos sobre la ontología. Esto es por el hecho de que el API de JessTab tiene una limitación tecnológica ya que es funcional pero no escalable.

Por tanto, se procedió a la reingeniería del código fuente de las funciones en cuestión obteniendo los resultados de la la 3.18.

Caso de prueba		Resultados	Observaciones
Inicio de sesión anónima	Caso 1	User has started anonymous session: 412A788AD70B2C05AFB5D510D5152D84	No existen observaciones.
	Caso 2	User has started anonymous session: DC045C8AD837E6373DDF20722EBFA2C0	No existen observaciones.
	Caso 3	User has started anonymous session: B62DDD199A98E29BE314B797A647D484	No existen observaciones.
Ingresar al Agente	Caso 1	Login 412A788AD70B2C05AFB5D510D5152D84 usuario01	No existen observaciones.
	Caso 2	Login DC045C8AD837E6373DDF20722EBFA2C0 usuario02	No existen observaciones.
	Caso 3	Invalid user name or password.	Se ha ingresado una contraseña inválida por tanto no se reconoce al usuario.
Salir del Agente	Caso 1	Logout usuario01	No existen observaciones.
	Caso 2	Logout usuario02	No existen observaciones.
	Caso 3	Logout usuario03	No existen observaciones.
Recomendación de partes o accesorios	Caso 1	productRecommendReturn (long): 20	No existen observaciones.
	Caso 2	productRecommendReturn (long): 60 63	No existen observaciones.
	Caso 3	productRecommendReturn (long): 1163 1165 1001	No existen observaciones.
Recomendación de	Caso 1	productRecommendReturn (long): 243 248	No existen observaciones.

productos de la categoría	Caso 2	productRecommendReturn (long): 231 251	No existen observaciones.
	Caso 3	productRecommendReturn (long): 87 82 207	No existen observaciones.
Recomendación de productos de interés	Caso 1	productRecommendReturn (long): 229 59 247	No existen observaciones.
	Caso 2	productRecommendReturn (long): 168 171 222 227 221	No existen observaciones.
	Caso 3	productRecommendReturn (long): 189 241 42 127 142 210 170	No existen observaciones.
Recomendación de promociones, publicidad o servicios asociados a producto, partes o accesorios	Caso 1	No se recibe ningún resultado.	El servicio de recomendación no responde. Se observa que el porcentaje de uso de la CPU sube al 100%. Se procede cancelar el proceso.
	Caso 2	No se recibe ningún resultado.	Idem.
	Caso 3	No se recibe ningún resultado.	Idem.
Recomendación de de promociones, publicidad o servicios asociados a la categoría	Caso 1	No se recibe ningún resultado.	Idem.
	Caso 2	No se recibe ningún resultado.	Idem.
	Caso 3	No se recibe ningún resultado.	Idem.
Recomendación de promociones, publicidad o servicios de interés	Caso 1	No se recibe ningún resultado.	Idem.
	Caso 2	No se recibe ningún resultado.	Idem.
	Caso 3	No se recibe ningún resultado.	Idem.

Tabla 3.16 Resultados de las pruebas (Los Autores)

Caso de prueba		Resultados	Observaciones
Recomendación de promociones, publicidad o servicios asociados a producto, partes o accesorios	Caso 1	promotionRecommendReturn (long): 2	No existen observaciones.
	Caso 2	adRecommendReturn (long): 3 1	No existen observaciones.
	Caso 3	serviceRecommendReturn (long): 2 4 5 1 3	No existen observaciones.
Recomendación de de promociones, publicidad o servicios asociados a la categoría	Caso 1	promotionRecommendReturn (long): 6 7 5	No existen observaciones.
	Caso 2	adRecommendReturn (long): 6 7 5	No existen observaciones.
	Caso 3	serviceRecommendReturn (long): 8 6	No existen observaciones.
Recomendación de promociones, publicidad o servicios de interés	Caso 1	promotionRecommendReturn (long): 2	No existen observaciones.
	Caso 2	adRecommendReturn (long): 8 4	No existen observaciones.
	Caso 3	serviceRecommendReturn (long): 9 7 3	No existen observaciones.

Tabla 3.17la 3.18 Resultados de las pruebas que fallaron la primera vez (Los Autores)

CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- La comunicación es un factor muy importante para que la estrategia marketing funcione con el mercado meta. Por este motivo, los elementos relacionados con la comunicación, por ejemplo publicidad e información, deben ser elaborados de manera metódica e inteligente. De esta forma, una empresa es capaz de cumplir efectivamente con los objetivos de su estrategia de mercadeo.
- La descripción del comportamiento del consumidor es un tema bastante complejo debido a la gran cantidad de variables que influyen en la decisión de compra. En la actualidad existen varios intentos que tratan de describir estructuradamente la manera de proceder de los clientes; lo que permite que los modelos de comportamiento del consumidor puedan ser representados a través de programas de software. De esta manera, los sistemas de información pueden ser capaces de interactuar estratégicamente con los usuarios y ofrecer contenido altamente personalizado, como lo es en el caso de las recomendaciones de productos y otros elementos del marketing.
- Las ontologías tienen un alto grado de eficacia para la representación de información compleja, ya que sintetizan varias nociones de otras técnicas como por ejemplo bases de datos, teoría de objetos, redes semánticas y lógica matemática. Por este motivo, las ontologías tienen un gran potencial de uso y aplicación en diversas áreas especialmente en el campo de la ingeniería de software.
- Las ontologías constituyen un paradigma de representación del conocimiento que hasta la actualidad no pretende reemplazar la teoría de objetos ni las bases de datos. Sin embargo, la utilización combinada de estas técnicas potencia significativamente la funcionalidad de software debido a sus variadas aplicaciones.
- Dentro del área de e-commerce, las ontologías tienen un amplio grado de aplicación al permitir la representación de conocimiento por parte de expertos como por ejemplo gerentes de ventas o marketing. De esta manera se puede

utilizar en los sistemas de software información más compleja acerca de productos, clientes y otros aspectos del negocio.

- Los agentes inteligentes aportan al nacimiento de una nueva generación de software que permite la realización de tareas más complejas sin intervención humana. Además, los agentes inteligentes van a cumplir un papel fundamental dentro de la Web Semántica al participar en las diferentes alternativas de comunicación que se presentan entre usuarios y sistemas.
- En el campo del comercio electrónico, los agentes inteligentes tienen un alto grado de aplicación al permitir que varias tareas complejas sean delegadas por usuarios humanos como por ejemplo tareas de búsqueda de productos, tareas de subasta y tareas de venta. De esta manera se logra incrementar el nivel de automatización de los negocios electrónicos.
- Las ontologías, al representar formalmente el conocimiento, constituyen un excelente protocolo de comunicación entre sistemas multi-agente con lo cual se puede incrementar el nivel de interacción y aprendizaje de los mismos.
- La aplicación de ontologías y agentes inteligentes dentro del campo del comercio electrónico ofrece un gran incremento en el nivel de funcionalidad de los sistemas de e-commerce ya que surge una gran cantidad de nuevas alternativas que se pueden aplicar al automatizar ciertos aspectos de la estrategia del negocio.

4.2. RECOMENDACIONES

- A pesar de que UML es un lenguaje exclusivo para representación de objetos, puede ser adaptado para la representación de ontologías; sin embargo, es necesario resaltar que tiene varias deficiencias especialmente en la representación de restricciones de los slots. De la misma manera, UML resulta en parte adecuado para representar varios elementos de un sistema basado en reglas pero también existen deficiencias para la representación de algunos detalles. Es recomendable hacer varias extensiones a este lenguaje de tal modo que facilite los procesos de análisis y diseño al utilizar estos dos tipos de paradigmas.
- El RUP es una metodología que tiene muchas intuiciones referentes a las relaciones semánticas las cuales son completamente explícitas en el campo

de la ingeniería ontológica. Por este motivo, al tener presente un enfoque ontológico en el desarrollo de modelos con UML, algunas actividades que especifica el RUP se vuelven redundantes. Por tanto, algunas tareas pueden ser perfectamente omitidas o simplificadas para reducir el tiempo de análisis y diseño. Es recomendable hacer un análisis de los beneficios que se pueden obtener al aplicar los principios de la ingeniería ontológica con esta metodología.

- Si bien el RUP es una metodología para desarrollo de sistemas orientados a objetos tiene algunas características que pueden ser adaptadas para la creación de agentes de software simples; sin embargo, es necesario anotar que existen algunas limitaciones que afectan el desarrollo de los mismos. Es recomendable hacer extensiones al RUP para el desarrollo de agentes de software o en su defecto utilizar otra metodología que sea más acorde para esta clase de sistemas de software.
- Dentro del paradigma ontológico existen algunas generalizaciones acerca de las clases que pertenecen a cierto dominio en particular. Al realizar la integración con el paradigma de reglas se pueden crear elementos como funciones, consultas y reglas acerca de las clases de nivel jerárquico superior de esta forma se logra un gran nivel de reutilización de código y se pueden reducir los tiempos de implementación de un sistema. Es recomendable analizar la serie de ventajas que se pueden tener al crear algoritmos para tareas genéricas que tengan una amplia variedad de aplicación en diversos dominios.
- El éxito de la implementación del APRS se dio ya que Protégé, JESS y AXIS están desarrollados sobre Java; lo que ha permitido una fácil integración de estas tres tecnologías a través del IDE de Eclipse. Se recomienda el uso de tecnología Java para la construcción de proyectos de software de similares características.
- Protégé y JESS son tecnologías que tienen orígenes que pertenecen a paradigmas computacionales diferentes. A pesar de que su integración fue exitosa, se pudo observar en la fase de pruebas que existen algunas limitaciones tecnológicas que influyen en el desempeño de la aplicación motivo por el cual se tuvo que hacer una reingeniería de código. Se recomienda

analizar las ventajas de otras plataformas de software que vienen ya con los dos tipos de paradigmas integrados. (Ej.: OpenCyc, KAON o JENA).

REFERENCIAS BIBLIOGRÁFICAS

LIBROS Y MANUALES

- [1]. BANDA, Hugo. Sistemas de Inteligencia Artificial. Escuela Politécnica Nacional. Facultad de Ingeniería de Sistemas. 1997.
- [2]. BIGUS, Joseph; BIGUS Jennifer. Constructing Intelligent Agents Using Java. Second Edition. Willey. New York. 2001.
- [3]. CLIPS. Reference Manual. Volume I. Basic Programming Guide. Version 6.23. Ene. 2005.
- [4]. DALRYMPLE, Douglas; PARSONS, Leonard. Marketing Management: Strategy and Cases. Willey. New York. 1990.
- [5]. EELES, Peter; et al. Building J2EE Applications with the Rational Unified Process. Addison Wesley. 2003.
- [6]. ERICSSON, Henrik. JessTab Manual: Integration of Protégé and Jess. Linköping University. Ene. 2004.
- [7]. FIEDMAN-HILL, Ernest. JESS in Action: Rule-Based Systems in Java. Manning. Greenwich. 2003.
- [8]. IMHOFF, Claudia; GENTRY, Jeff. Customer Relationship Management: Building an Environment for the Future. Jerry Ryan (Ed.) The Technology Guide Series.
- [9]. KLUSH, Matthias. Intelligent Information Agents, Agents-Based Information Discovery and Management on the Internet. Springer. Berlin. 1999.
- [10]. KOTLER, Philip. Dirección de Marketing. Prentice-Hall. 2000.
- [11]. MICROSOFT PRESS. Developing XML Web Services and Components with Microsoft Visual Basic.NET and Visual C#.NET. Redmon, Washington. 2003.
- [12]. NOY, Natalya F.; MCGUINNESS, Deborah L. Ontology Development 101: A Guide to Creating Your First Ontology. Stanford University. Stanford, CA. 2000.
- [13]. ROSENTAL, M.; IUDIN, P. Diccionario Filosófico. Ediciones Nacionales. Bogotá.
- [14]. RUSSELL, Stuart; NORVIG, Meter. Inteligencia Artificial: Un enfoque moderno. Prentice Hall Hispanoamericana. México. 1996.

- [15]. SCHMULLER, Joseph. Aprendiendo UML en 24 horas. Prentice Hall.
- [16]. STANFORD UNIVERSITY. Protégé-2000 User Guide. Stanford, CA. 2000.

ARTÍCULOS

- [17]. CTA. What Ontologies Mean to Your Business. 2004.
- [18]. ECOMMERCE INNOVATION CENTRE. Ecommerce Case Study. Canon Hill Potteries Ltd. Cardiff University. Feb. 1999.
- [19]. GARSHOL, Lars Marius. Metadata? Thesauri? Taxonomies? Topic Maps! Making sense of it all. Journal of Information Science. pages 378-391, volume 30, number 4. Oct. 2004
- [20]. GONÇALVES, José; DE ALMEIDA, Ricardo. Análise de Domínio e Ontologias. Dissertação de Mestrado, Universidade Federal do Espírito Santo, Departamento de Informática. 2004.
- [21]. GRUBER, Thomas. A Translation Approach to Portable Ontology Specifications Knowledge Acquisition. California. Abr. 1993.
- [22]. GRUBER, Thomas. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. California. Ag. 1993.
- [23]. JIMENEZ, Silvestre; Ramos Esmeralda. Agentes Inteligentes. Lecturas en Ciencias de la Computación. Universidad Central de Venezuela, Facultad de Ciencias, Escuela de Computación. Caracas, Abr. 2000.
- [24]. JULIÁN, V.; BOTTI, V. Agentes Inteligentes: El siguiente paso en la inteligencia artificial. Dpto. Sistemas Informáticos y Computación, Universidad Politécnica de Valencia. Valencia, España, May.-Jun. 2000.
- [25]. LOZANO, Adolfo. Ontologías en la Web Semántica. I Jornadas de Ingeniería Web' 01. España. 2001.
- [26]. MIZOGUCHI, Riichiro. Ontology development, tools and languages.
- [27]. PIVK, A. E-Commerce Intelligent Agents. Department of Intelligent Systems, Jozef Stefan Institute. Ljubljana, Slovenia. 2001.
- [28]. SHAFER, J. B; et al. E-Commerce Recommendation Applications. GroupLens Research Project, Department of Computer Science and Engineering, University of Minnesota. Minneapolis, MN. 2000.
- [29]. SMITH, Barry. Ontology. Draft version of chapter published in Luciano Floridi (ed.), Blackwell Guide to the Philosophy of Computing and

Information, 155-166. Oxford. 2003

- [30]. ZÚÑIGA Gloria. *Ontology: Its Transformation From Philosophy to Information Systems*. The Acton Institute. Oct. 2001.

DIRECCIONES ELECTRÓNICAS

- [31]. ALFAWORKS. What is ontology? Frequently asked questions. IBM. www.alphaworks.ibm.com/contentnr/semanticsfaqs [Consulta Oct. 2005].
- [32]. APACHE. Apache Tomcat tomcat.apache.org [Consulta: Abr. 2006]
- [33]. AXIS. Apache Web Services Project ws.apache.org/axis [Consulta Abr. 2006]
- [34]. CANGELOSI, Joe. *Consumer Behavior: Marketing*, Capítulo 6. [Consulta: May. 2005]. Presentación de diapositivas.
- [35]. CHAFFEY, Dave. *Internet Marketing Book*. www.marketing-online.co.uk/ch1.pdf [Consulta: May. 2005]
- [36]. ECLIPSE. The Eclipse Project. www.eclipse.org. [Consulta: Abr. 2006]
- [37]. FORMAL ONTOLOGY. What is Ontology? Definitions by leading philosophers from Christian Wolff to Edmund Husserl. www.formalontology.it/section_4.htm Sept. 2004
- [38]. JESS. The Rule Engine for the Java Platform. herzberg.ca.sandia.gov/jess [Consulta Abr. 2006]
- [39]. JESSTAB. Integrating Protégé and Jess. www.ida.liu.se/~her/JessTab [Consulta: Abr. 2006]
- [40]. MUÑIZ, Rafael. *Márketing en el Siglo XXI*. www.marketing-xxi.com [Consulta: Jun. 2005]
- [41]. OPENCYC. Formalized Common Knowledge. www.opencyc.org [Consulta: Jun 2005]
- [42]. PRESSLEY, Milton. *Analyzing Consumer Markets and Buyer Behavior*. Prentice Hall. 2003. Presentación de diapositivas.
- [43]. PROTÉGÉ. Stanford Medical Informatics. Stanford University School of Medicine. protege.stanford.edu [Consulta: Jun. 2005]
- [44]. PSMTAB. The PSM Librarian tab plugin for Protégé. protege.stanford.edu/plugins/psmtab/PSMTab.html [Consulta: Jun. 2005]
- [45]. RICHTER, Michael. *Knowledge Management in e-Commerce*. sern.ucalgary.ca/courses/SENG/609.13/ Nov. 2003.

- [46]. ROSSETANET. www.rosettanet.org [Consulta: Jun. 2005]
- [47]. SUN MICROSYSTEMS. Java 2 Platform, Enterprise Edition java.sun.com. [Consulta: Abr. 2006]
- [48]. SUO. Standard Upper Ontology Working Group. suo.ieee.org [Consulta: Jun. 2005]
- [49]. THOMSON LEARNING. Marketing Management. South Western. Jun. 2005. Presentación de diapositivas.
- [50]. UMLS. Unified Medical Language System. U.S. National Library of Medicine www.nlm.nih.gov/research/umls/umlsmain.html [Consulta: Jun. 2005]
- [51]. UNSPSC. The United Nations Standard Products and Services Code. www.unspsc.org [Consulta: Jun. 2005]
- [52]. WIKIPEDIA. The free encyclopedia. en.wikipedia.org. [Consulta: Oct. 2005]
- [53]. WIKIPEDIA. La enciclopedia libre. es.wikipedia.org. [Consulta: Oct. 2006]

ACRÓNIMOS

API	Application Programming Interface
ARPS	Agente de Recomendación de Productos o Servicios basado en Ontologías
BDD	Base de datos
CLIPS	C Language Integrated Production System
CRM	Custom Relationship Management
DAML	DARPA Agent Markup Language
DTD	Document Type Definition
e-CRM	Electronical Custom Relationship Management
ERS	Especificación de Requerimientos de Software
IDE	Integrated Development Environment
IP	Internet Protocol
J2EE	Java 2 Platform, Enterprise Edition
JDT	Java Development Kit
JESS	Java Expert Shell System
JSP	Java Server Pages
JVM	Java Virtual Machine
ODBC	Open Data Base Connectivity
OIL	Ontology Inference Layer
OWL	Ontology Web Language
OWL DL	Ontology Web Language Description Logic

PAMA	Percepciones, Acciones, Metas y Ambiente
PSM	Problem Solving Method
RDF	Resource Description Framework
RDF-S	Resource Description Framework Schema
RUP	Rational Unified Process
SOAP	Simple Object Access Protocol
SUO	Standard Upper Ontology
UML	Unified Modeling Language
UNSPSC	Universal Standard Product and Services Classification
URI	Universal Resource Identifier
URL	Uniform Resource Locator
WSDL	Web Service Description Language
WTP	Web Tools Project
WWW	World Wide Web
XML	eXtensible Markup Language

GLOSARIO

- **Abstracción clave:** Clase de análisis que representa una entidad del negocio.
- **Base de conocimiento:** Conjunto de instancias de clases y sus definiciones que pueden ser utilizadas por un PSM
- **CLIPS:** Es una herramienta de desarrollo productivo para sistemas expertos que provee un ambiente completo para la construcción de sistemas basados en objetos o reglas. Fue creado en 1985, y actualmente es ampliamente utilizado académica e industrialmente debido a sus características.
- **Conceptuación:** Acción de conceptuar. Conceptuar: Formar concepto de una persona o cosa.
- **Front Controller:** Es un patrón de diseño J2EE para una lógica centralizada de peticiones. Puede ayudar a eliminar código duplicado en una serie de peticiones al redirigirlas a través de controlador frontal.
- **Hecho:** En un sistema basado en reglas: pieza de información con que trabajan las reglas y conforman la memoria de trabajo.
- **Lógica de descripción:** Es una familia de lenguajes de representación del conocimiento que se utilizan para representar conocimiento terminológico de un dominio de aplicación de una manera formal, estructurada y comprensible. El nombre de lógica de descripción se refiere a las descripciones de conceptos que se utilizan para describir un dominio y además a su semántica basada en lógica que puede ser trasladada a una lógica de primer orden. La lógica de descripción fue diseñada como una extensión de plantillas y redes semánticas las cuales no están equipadas con una semántica basada en lógica formal.
- **Lógica de primer orden:** Es un tipo de lógica matemática que permite la formulación de proposiciones cuantificadas como por ejemplo “Existe una x tal que...” ($\exists x$) o “Para todas las x...” ($\forall x$), en donde x es un miembro de un dominio de discurso.
- **Mapear:** (Anglicismo). Relacionar conceptos lógicamente, de acuerdo a su

significado.

- **Mecanismo de análisis:** Es una capacidad arquitectónica que representa una solución común a un problema común. Los mecanismos representan capacidades generales que soportan la funcionalidad básica del sistema.
- **Mecanismo de diseño:** Patrón de diseño, servicio o elemento o que pueden ser utilizados varias veces en una aplicación a nivel de diseño.
- **Mecanismo de implementación:** Tecnología, servicio o elemento que pueden ser utilizados varias veces en la implementación de una aplicación.
- **Ontología:** Un modelo de un campo del conocimiento en particular, los conceptos y sus propiedades así como también las relaciones entre los conceptos.
- **Plantilla:** Estructura de datos apropiada para representar una situación estereotípica. Las plantillas organizan el conocimiento en objetos y eventos que resultan apropiados para situaciones específicas.
- **Plantilla de hechos:** En un sistema basado en reglas. Estructura de datos que define slots para almacenamiento de hechos.
- **Problem Solving Method (PSM):** Método de resolución de problemas es programa de computadora que se usa junto con una base de conocimiento para responder preguntas y resolver problemas.
- **Proxy:** Un objeto que habilita la comunicación entre procesos.
- **Refinar:** Modificar un componente del conocimiento para obtener una nueva versión de éste y sirva para su uso práctico.
- **Servicio Web:** Sistema de software diseñado para proveer una interfaz que puede ser usada por otro software. El término servicio Web generalmente se refiere a una aplicación que puede enviar comandos utilizando el protocolo simple de accesos a objetos SOAP que está basado en XML. Los servicios Web puede ser usados para construir aplicaciones Web con una interfaz de usuario regular; sin embargo son más utilizados como componentes sistemas de software más grandes.
- **Servlet:** Elemento definido en J2EE que consiste en un pequeño módulo que es invocado por un servidor Web u otro contenedor en respuesta a la petición de un navegador de Internet.

- **Simple Object Access Protocol (SOAP):** Protocolo simple de acceso a objetos. Permite el intercambio estructurado y con tipos de datos de información entre aplicaciones sobre el Internet.
- **Sistema basado en conocimiento:** Sistema de computación que tiene una base de conocimiento acerca de un dominio y programas que incluyen reglas para el procesamiento del conocimiento y para resolver problemas relacionados al dominio.
- **Slot:** Propiedad de una clase. Por ejemplo la clase persona puede tener nombre, título, y número telefónico como slots.
- **Web Service Description Language (WSDL):** Lenguaje de descripción de servicios Web. Especificación en formato XML que contiene información acerca de las capacidades de un servicio Web, sus localización y cómo interactuar con éste.