

NEGOCIACIÓN AUTOMÁTICA EN MERCADOS ELECTRÓNICOS

**MÁSTER EN INVESTIGACIÓN
EN INFORMÁTICA
2007/2008**

TRABAJO DE INVESTIGACIÓN



Departamento de Ingeniería de Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid
MADRID - ESPAÑA

**Alumna:
Carla Salazar Serrudo**

**Director:
Luis Javier García Villalba**

Resumen

Existen varias etapas al tiempo de comprar productos en aplicaciones de comercio electrónico. La etapa más interesante es el proceso de negociación. Este proceso se considera el momento más importante en la compra y venta de productos, pero también es una tarea muy compleja y que requiere de tiempo y pericia para conseguir un buen precio, tanto del comprador como del vendedor.

El presente estudio responde a un objetivo principal, el cual es investigar el mercado electrónico de los negocios y sus características, además de identificar el algoritmo de Aprendizaje Automático aplicable al proceso de negociación electrónica.

En este contexto y basados en el trabajo de Kasbah [7], en este proyecto se ha desarrollado un prototipo de Sistema de Aprendizaje de Negociación Electrónica (SANE) donde existe un mercado electrónico en el que un agente comprador y un agente vendedor negocian el precio de un producto para conseguir el “mejor precio posible”. Los agentes se comunican usando un lenguaje simbólico de negociación diseñado para este fin e incluyen parámetros de tiempo de negociación y cantidad dinero como restricciones a la hora de negociar. El aprendizaje del agente comprador se concreta a través del Razonamiento Basado en Casos, para que pueda aprender de experiencias pasadas cómo calcular el nuevo valor a ofertar y conseguir el mejor precio.

Los resultados obtenidos son alentadores porque se demuestra que los agentes software que incluyen Razonamiento Basado en Casos negocian de mejor forma que los no la tienen. Aunque aún quedan muchas tareas que realizar, se cree que en un futuro cercano se puede conseguir implantar la negociación automática en el comercio electrónico Business to Consumer.

Palabras clave: agentes software, comercio electrónico, negociación automática, aprendizaje automático, razonamiento basado en casos.

Agradecimientos

Este Proyecto se ha realizado con el apoyo del Programa Alban, Programa de Becas de Alto Nivel de la Unión Europea para América Latina, beca nº E07D401204BO.

A mi tutor, Dr. Luis Javier García Villalba, por todo la ayuda brindada en la realización de este Proyecto.

Y en especial, a Jorge Lizandro Coca Olmos, que me brinda su apoyo incondicional en todo momento y situación.

A mi hijo Rafael Arturo

Tabla de Contenidos

1. Introducción	7
1.1 Antecedentes.....	7
1.2 Identificación del Problema.....	8
1.3 Objetivos de la Investigación.....	9
1.3.1 Objetivos Generales	9
1.3.2 Objetivos Específicos.....	9
1.4 Metodología a Seguir.....	10
1.5 Aportaciones de la Investigación.....	11
2. Algoritmos de Aprendizaje Automático	12
2.1 Conceptos Generales	12
2.2 Revisión de Diferentes Técnicas de Aprendizaje Automático.....	12
2.2.1 Árboles de Decisión.....	13
2.2.2 Reglas de Decisión.....	15
2.2.3 Programación Lógica Inductiva	17
2.2.4 Redes Neuronales	20
2.2.5 Aprendizaje No-Supervisado	23
2.2.6 Aprendizaje por Refuerzo	25
2.2.7 Razonamiento Basado en Casos.....	27
3. Comercio Electrónico Basado en Agentes	29
3.1 Introducción.....	29
3.2 Infraestructura para el Comercio Electrónico	30
3.3 Etapas del Comercio Electrónico	32
3.4 Proceso de Negociación	34
3.4.1 Definición de Negociación.....	35
3.4.2 Zona de Acuerdo	35
3.4.3 Estructura para Negociación Electrónica.....	37
3.4.4 Desempeño de la Negociación.....	38
3.4.5 Sistemas de Información de Apoyo a los Procesos de Negociación	39
4. Desarrollo del Sistema de Negociación Automática	41
4.1 Introducción.....	41
4.2 Trabajos Relacionados	41
4.3 Propuesta de Negociación Automática	43
4.4 Prototipo de Sistema de Negociación Electrónica	44
4.4.1 Agentes Software	45
4.4.2 Lenguaje de Comunicación	50
4.4.3 Pizarra de Comunicación	51
4.4.4 El Mercado Electrónico	52
4.5 Diseño del Sistema de Negociación Electrónica	55
4.6 Ejemplo de Funcionamiento del Sistema de Negociación Electrónica	56
4.7 Análisis de Resultados.....	60

5. Negociación Automática con Aprendizaje	63
5.1 Introducción	63
5.2 Análisis de los Algoritmos de Aprendizaje	63
5.3 Arquitectura del Sistema de Aprendizaje de Negociación Electrónica	64
5.3.1 Agente Comprador Inteligente	65
5.3.2 Casos de SANE	66
5.3.3 Memoria de Casos	68
5.3.4 Gestión de Casos	69
5.4 Diseño del Sistema de Aprendizaje de Negociación Electrónica	72
5.5 Funcionamiento del Sistema de Aprendizaje de Negociación Electrónica	73
5.6 Análisis de Resultados	76
6. Conclusiones y Trabajo Futuro	79
6.1 Conclusiones	79
6.2 Trabajo Futuro	81
6.3 Divulgación de Resultados	82
Bibliografía	83
Anexo A	88
Artículo: “ <i>Comercio Electrónico B2C Personalizado</i> ”	88

1. Introducción

1.1 Antecedentes

Con el extraordinario desarrollo de Internet y World Wide Web (WWW), el comercio electrónico ha captado la atención en los últimos años, porque plantea un nuevo marco de negocios para desarrollar operaciones mercantiles. Se puede definir al comercio electrónico como cualquier forma de transacción comercial en el que las partes interesadas interactúan electrónicamente, en lugar de por intercambio o contacto físico directo [8][44]. De esta forma, las actividades tradicionales como la compra, la intermediación, la negociación y la venta de productos se han transformado en las actuales e-compra, e-intermediación, e-negociación y e-venta.

Se puede definir, de manera general, estas actividades así:

- **Compra** (en inglés *shopping*): Generalmente se refiere al acto de comprar productos, pero también puede consistir en sólo observar los productos de las tiendas [46].
- **Intermediación** (del inglés *brokerage*): Es la parte que media entre el comprador y el vendedor. Una persona o empresa que compra o vende para el cliente.
- **Negociación** (del inglés *negotiation*): Consiste en una interacción de influencias entre las partes con miras a alcanzar la adquisición de un producto o servicio. Esta interacción incluye, por ejemplo, la resolución de disputas, logro de acuerdos, regateo de precio y las condiciones de adquisición del producto.
- **Venta** (del inglés *retailing*): Consiste en la venta de productos a través de una tienda.

Las actividades del comercio electrónico han propiciado el desarrollo de aplicaciones industriales y de investigación. Existen dos formas de aplicaciones de comercio electrónico: la primera consiste en poner productos y servicios en línea de la misma

forma que la tradicional y la segunda que usa técnicas de aprendizaje automático para efectuar una transformación inteligente y conseguir éxito en la compra y venta de productos [47]. La intensa competición entre negocios electrónicos para intentar conseguir nuevos clientes y retener a los existentes ha hecho que las técnicas inteligentes sean indispensables para el comercio electrónico.

1.2 Identificación del Problema

Según Guttman, Moukas y Maes [16], las etapas a la hora de comprar algún producto son las siguientes:

1. Identificación de las necesidades del comprador
2. Elección del producto a comprar
3. Elección del vendedor
4. Negociación
5. Compra y entrega
6. Uso y evaluación del producto adquirido.

Se puede decir que la etapa más interesante es el proceso de negociación [44]. Este proceso se considera el momento más importante dentro del comercio, pero a su vez, es una tarea bastante compleja y que requiere de tiempo y de estrategias, tanto del comprador como del vendedor para conseguir el producto en las mejores condiciones posibles.

El proceso de negociación consiste en que las distintas partes interesadas buscarán la aceptación de sus ideas, intereses o propuestas, siendo su objetivo común conseguir un resultado consensuado que satisfaga la expectativa de todas las partes involucradas [11]. El proceso de negociación de un producto puede darse por diferentes criterios: el precio, la cantidad de artículos, el plazo de entrega, las condiciones de entrega y pago, etc. En general, el regateo se relaciona con la negociación por precio y en este proyecto nos abocaremos a este tema.

1.3 Objetivos de la Investigación

La automatización de la negociación ha recibido gran atención de parte de la comunidad de investigadores de sistemas multiagentes, por sus bondades a la hora de reducir el tiempo de negociación y eliminar algunas de las desventajas de la negociación humana tales como: el sentimiento embarazoso de pedir rebajas, de perder mucho tiempo, de sentirse desorientado, etc.

1.3.1 Objetivos Generales

- Investigar acerca del comercio electrónico y sus características en la literatura relevante.
- Identificar los algoritmos de aprendizaje automático aplicables al proceso de negociación en mercados virtuales.
- Analizar las ventajas e inconvenientes de cada uno de estos algoritmos.
- Implementar un mercado electrónico, donde participen agentes software, que hagan las veces de compradores y vendedores.
- Implementar el proceso de negociación del precio de un producto entre agentes vendedores y compradores.
- Implementar el aprendizaje del proceso de negociación electrónica, aplicando el algoritmo elegido.
- Verificar y validar los resultados obtenidos con el proceso de aprendizaje versus el proceso de negociación sin aprendizaje.

1.3.2 Objetivos Específicos

- Determinar las características del mercado electrónico a implementar.
- Determinar las características que deberán incluir los agentes compradores y vendedores.
- Seleccionar el algoritmo de aprendizaje automático conveniente.

- Aplicar el algoritmo de aprendizaje seleccionado al proceso de negociación electrónica.
- Construir los casos ejemplos.
- Realizar los experimentos pertinentes para verificar y validar el proceso de negociación, con y sin aprendizaje.

1.4 Metodología a Seguir

Para la realización del presente proyecto se seguirán los siguientes pasos:

- a) Recopilación de las principales fuentes teóricas acerca de las áreas en cuestión: Comercio Electrónico, Agentes Software y Aprendizaje Automático, ya sea en forma de libros o artículos científicos.
- b) Investigación de las últimas soluciones para la construcción de agentes software.
- c) Conceptualización del sistema a construir:
 - Identificación del tipo de mercado electrónico a implementar.
 - Determinación de las características y comportamiento de los agentes software a desarrollar.
 - Especificación del proceso de negociación que se llevará a cabo.
- d) Implementación del mercado electrónico donde los agentes compradores y vendedores podrán negociar el precio de un producto.
- e) Verificación y validación del funcionamiento del mercado electrónico y del proceso de negociación.
- f) Evaluación comparativa de las diferentes técnicas de Aprendizaje Automático para implantarlo en los agentes software negociantes.
- g) Implementación de la técnica de Aprendizaje Automático elegida en los agentes.
- h) Verificación y validación del algoritmo de aprendizaje de procesos de negociación implementado.
- i) Elaboración del documento técnico.

1.5 Aportaciones de la Investigación

Con la realización de este proyecto se pretende dar un paso importante en la Automatización de la Negociación en los mercados electrónicos, puesto que se pretende conseguir agentes software inteligentes a los cuales se les pueda delegar la tarea de negociar el precio de un producto. Dichos agentes deberían aprender gradualmente, de los usuarios y de otros agentes similares, cómo regatear el precio de un producto en aplicaciones de comercio electrónico. Se puede mencionar que, los trabajos realizados actualmente en el área de Negociación Automática son bastante teóricos y siempre apuntan a conseguir el valor óptimo con estrategias que tienen complejidad NP. En cambio, en esta aproximación, queremos construir agentes que encuentren el mejor valor posible, aunque no sea el óptimo, pero que pueda funcionar de manera práctica en las aplicaciones Business to Consumer (B2C) en línea.

2. Algoritmos de Aprendizaje Automático

2.1 Conceptos Generales

El Aprendizaje Automático es un área de la Inteligencia Artificial y está relacionada con el diseño y desarrollo de algoritmos y técnicas que permitan a la computadora a “aprender” [5]. En un nivel general, existen dos tipos de aprendizaje: inductivo y deductivo. El aprendizaje inductivo es el más popular y consiste en extraer reglas y patrones de un conjunto grande de datos.

El objetivo del Aprendizaje Automático es conseguir información automáticamente a partir del conjunto de datos, mediante el uso de métodos computacionales y estadísticos. Por esta razón, el Aprendizaje Automático está muy relacionado con diferentes áreas de las Ciencias de la Computación, como son: la Minería de Datos, la Estadística y los Modelos Computacionales Teóricos.

El Aprendizaje Automático se aplica a varios campos del saber incluyendo al procesamiento de lenguaje natural, reconocedor de patrones, motor de búsqueda, diagnóstico médico, bio-informática, quim-informática, detección de fraudes por tarjeta de crédito, análisis de mercados, clasificación de secuencias de DNA, reconocedor de voz y de escritura, juegos y muchos más.

2.2 Revisión de Diferentes Técnicas de Aprendizaje Automático

En este apartado se describirán varias técnicas de Aprendizaje Automático clasificados por la estructura de lo que se aprenderá. Se han incluido las que se consideran pertinentes al área del proyecto.

2.2.1 Árboles de Decisión

ID3 y C4.5 son algoritmos creados por Ross Quinlan [3] [32] en 1986 y en 1993, respectivamente. El objetivo de ambos algoritmos es inducir árboles de decisión a partir de un conjunto de ejemplos. Se debe disponer de un conjunto de ejemplos con la misma estructura. Los ejemplos consisten en pares atributo-valor. Uno de estos atributos representa la categoría del ejemplo. El problema consiste en determinar un árbol de decisión compuesto de los atributos no categóricos que clasifique a los ejemplos en algún valor del atributo categórico.

Por ejemplo, se dispone de un conjunto de ejemplos que representan a personas y se tienen datos referentes a su altura, color de cabello y color de ojos. Estas personas están categorizadas por si pertenecen al curso de modelaje (+) o no pertenecen al curso de modelaje (-). Ver Tabla 1.

#	Altura	Cabello	Ojos	Categoría
1	Bajo	Rubio	Verdes	+
2	Alto	Rubio	Oscuros	-
3	Alto	Castaño	Verdes	+
4	Bajo	Negro	Verdes	-
5	Alto	Negro	Verdes	-
6	Alto	Rubio	Verdes	+
7	Alto	Negro	Oscuros	-
8	Bajo	Rubio	Oscuros	-

Tabla 1. Ejemplo para ID3. Ejemplo extraído de [14]

El resultado del algoritmo ID3 propuesto por Quinlan produce el árbol de decisión mostrado en la Figura 1, que resume el conjunto de datos. Este árbol también sirve para clasificar casos nuevos.

En el árbol, cada nodo corresponde a un atributo y cada rama un posible valor del atributo. La ruta desde la raíz del árbol hasta una hoja describe los atributos de un ejemplo. La hoja del árbol especifica el valor esperado del atributo de clasificación. En la

Figura 1 también se puede ver que dicho árbol ha clasificado los ejemplos, tomando en cuenta los diferentes atributos no-categoricos propuestos, por ejemplo: si una persona tiene el cabello de color negro, entonces no pertenece al curso de modelaje.

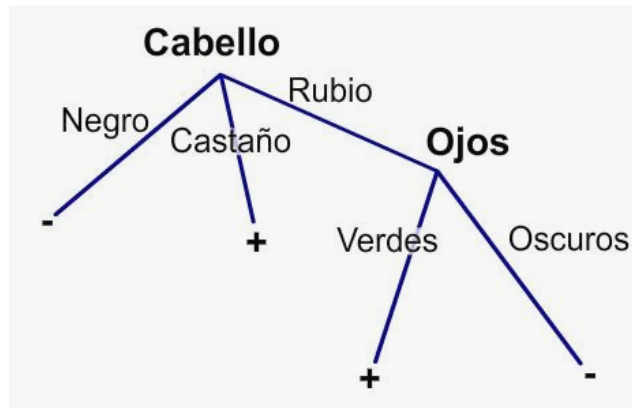


Figura 1. Ejemplo de árbol de decisión. Extraído de [14]

En un árbol de decisión, cada nodo debería estar asociado con el atributo que contenga “más información”, de entre los atributos aún no considerados. El concepto de *entropía* se usa para medir cuán informativo es un nodo y está asociado al concepto de *ganancia*. La *ganancia* ayuda a determinar qué atributo clasifica mejor los datos en cada nodo del árbol [3].

Dicho árbol, fácilmente se puede traducir en un conjunto de reglas de decisión, de la siguiente forma:

- *Si (Cabello = Negro) Entonces “-”*
- *Si (Cabello = Castaño) Entonces “+”*
- *Si (Cabello = Rubio) \wedge (Ojos =Verdes) Entonces “+”*
- *Si (Cabello = Rubio) \wedge (Ojos = Oscuros) Entonces “-”*

El algoritmo ID3 usa la estrategia “divide y conquistarás” para clasificar los ejemplos y aplica el concepto de entropía para valorar los atributos, lo que le permite construir árboles pequeños y sencillos. Cuando existen muchos datos y muchos atributos, ID3 no

es muy útil, porque genera árboles frondosos. Además, ID3 no generaliza y tampoco maneja ruido.

Por otra parte, el algoritmo C4.5 introduce un número de extensiones al algoritmo ID3. Al momento de construir un árbol de decisión pueden existir atributos con valor desconocido. Se puede tratar este problema estimando la probabilidad de los resultados posibles y clasificando los registros con estos valores calculados [22].

C4.5 también trata atributos con valores continuos de la siguiente manera. Supongamos que el atributo C_i tiene un rango continuo. Se examinan los valores para este atributo en el conjunto de entrenamiento y se los ordena así: A_1, A_2, \dots, A_m . Se particionan los registros entre algunos de estos valores, de manera que se consiga un número óptimo de particiones para el problema. Este procedimiento puede involucrar muchos cálculos [3] [32].

C4.5 también incorpora el concepto de poda de árboles muy complejos. Un árbol se poda cuando se reemplaza todo un subárbol por un nodo hoja. Se realiza el reemplazo si una regla de decisión establece que el error esperado en un subárbol es más grande que en una sola hoja [3].

2.2.2 Reglas de Decisión

Una regla de decisión sirve para caracterizar el concepto a aprender [22] [39]. Las reglas de decisión tienen la siguiente forma:

If A_1 and A_2 and ... and A_n then C

Donde C es el concepto a aprender y A_i es el par atributo-valor $A_i = V$ ó de manera general $A_i = V_1 \vee V_2 \vee V_3 \vee \dots$, donde un atributo puede tomar uno de varios valores.

La familia de algoritmos AQ fue creada por Michalski y Chilausky [5]. Es un sistema de aprendizaje que genera reglas de decisión a través del progresivo *cubrimiento* de los datos de entrenamiento. Esta aproximación ha sido implementada en toda una familia de métodos derivados del algoritmo que fue publicado por Michalski en 1969. El algoritmo AQ consiste en “buscar un conjunto de reglas de decisión que cubran todos los ejemplos positivos sin cubrir a los negativos” [14].

El objetivo del algoritmo AQ consiste en encontrar el conjunto mínimo de reglas de decisión que caractericen el concepto a aprender. La descripción de un concepto sigue la forma normal disyuntiva (FND):

$$C = (A_{11} \wedge A_{12} \wedge \dots) \vee (A_{21} \wedge A_{22} \wedge \dots) \vee \dots \vee (A_{m1} \wedge A_{m2} \wedge \dots)$$

Es decir, una disyunción de reglas de decisión. El número de \vee debe ser mínimo y el número de \wedge en cada paréntesis debe ser máximo. A esto se llaman “generalizaciones conjuntivas de máxima especificidad” (MSC-G). La MSC-G que cubre más objetos es la caracterización de los objetos típicos (menos \wedge). La MSC-G que cubre menos objetos es la caracterización de los objetos atípicos. También se debe mencionar que las reglas generadas por el algoritmo AQ se pueden intersectar, unir y ordenar lógicamente.

Michalski y Chilausky informaron de unos resultados espectaculares en la aplicación de AQ a un sistema experto para el diagnóstico de enfermedades en la planta de la soja. El número de enfermedades distintas (conceptos) era 15 y el de síntomas (atributos) 35. El sistema experto había sido construido con técnicas tradicionales de adquisición del conocimiento (entrevistas con expertos para la obtención de las reglas) y su precisión en el diagnóstico era de 71.8%. A lo largo del trabajo se habían llegado a recopilar 630 casos de enfermedades diagnosticadas, se tomaron al azar 290 (conjunto de entrenamiento) sobre los que se aplicó el algoritmo AQ. Las reglas obtenidas se comprobaron con los 340 casos restantes (conjunto de pruebas). La precisión obtenida al sustituir las reglas de los expertos por las inducidas fue del 97.6% [4].

Un ejemplo de resultado de la aplicación del algoritmo AQ para la descripción de las 15 enfermedades.

$$\begin{aligned} & ([hojas = normal] \wedge [tallo = anormal]) \vee \\ & ([úlceras = bajo_suelo] \wedge [color_úlceras = marrón] \dots) \vee \\ & \dots \\ & \rightarrow rhictomia_root_rot \end{aligned}$$

La familia de algoritmos AQ consigue reglas completas y consistentes. En dominios médicos ha tenido mucho éxito, generando reglas tan precisas como las de los expertos. Cuando el espacio de ejemplos es grande, es mejor que ID3. Las extensiones de AQ manejan ruido e imprecisión de los datos. Los sistemas derivados (Induce, AQ15, Poseidon) manejan lógica de predicados de primer orden, inducción constructiva, etc.

2.2.3 Programación Lógica Inductiva

Los sistemas de aprendizaje mencionados anteriormente usan un lenguaje de representación basado en el par atributo-valor para describir los objetos y las reglas de clasificación. Esta representación proposicional es simple y ha tenido mucho éxito. Por otra parte, los sistemas basados en lógica de primer orden son mucho más poderosos que los anteriores puesto que pueden representar descripciones complejas, estructuras de entidades y reglas de clasificación en lógica de primer orden. FOIL (First Order Inductive Learning) es un sistema de aprendizaje que combina las ideas de ambos grupos. Los objetos se describen usando relaciones, y a partir de éstas, FOIL genera reglas de clasificación expresadas en lógica de primer orden [33][34][35].

En la Figura 2 se puede ver un ejemplo de relaciones familiares o árbol genealógico. La lógica proposicional es muy útil, pero tiene sus limitaciones a la hora de representar conceptos. En este ejemplo, el concepto *padre* se debería representar de la siguiente forma:

padre12: padre 1 de 2

padre13: padre 1 de 3

...

lo que produce una representación completamente ineficiente e inflexible. En cambio, en lógica de primer orden se representaría así:

padre(X,Y): X es padre de Y

padre = {(1,2), (1,3), (3,4), (3,5), (3,6), (4,7)}

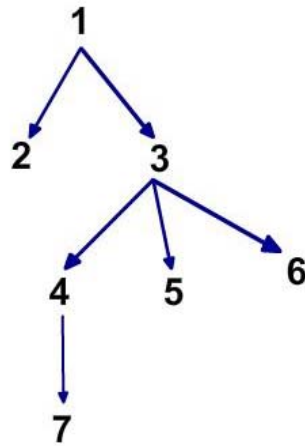


Figura 2. Ejemplo de árbol genealógico. Extraído de [35]

Si se desearía representar el concepto *abuelo* usando la lógica proposicional se debería crear el concepto *abuelo14* (abuelo 1 de 4) y así sucesivamente. En cambio, en lógica de primer orden se puede reutilizar el concepto *padre* para definir el concepto *abuelo*, tal como se verá más adelante.

abuelo(X, Y): X es abuelo de Y

abuelo(X, Y) = {(1,4), (1,5), (1,6), (3,7)}

Las cláusulas que se aprenderán en FOIL tendrán la siguiente forma:

$$C_1 :- L_{11}, L_{12}, \dots, L_{1m}$$

$$C_2 :- L_{21}, L_{22}, \dots, L_{2m}$$

...

donde el predicado C_i es la cabeza de la cláusula y los literales L_{ij} forman el cuerpo de la cláusula. Las comas indican conjunción. En AQ, el aprendiz empieza con una descripción general y poco a poco la especializa. FOIL sigue esa idea y empieza con la cabeza de la regla (C) y luego la va especializando añadiendo literales al cuerpo de la regla (L_{ij}). FOIL trabaja sobre relaciones definidas extensionalmente y construye definiciones generales.

En el ejemplo anterior, FOIL definirá el concepto *abuelo*, en una primera instancia, de la siguiente forma:

$$abuelo(X, Y) :- padre(X, Y).$$

Pero como dicha regla no es completa ni consistente, FOIL añadirá un nuevo literal a dicha regla, resultando de la siguiente manera:

$$abuelo(X, Y) :- padre(X, Z), padre(Z, Y).$$

FOIL también puede definir conceptos recursivamente. Así por ejemplo, el concepto *ancestro*:

ancestro(X,Y): X es ancestro de Y

$$ancestro(X, Y) = \{(1,2), (1,3), (1,4), (1,5), (1,6), (1,7), (3,4), (3,5), (3,6), (3,7), (4,7)\}$$

lo definiría con las dos siguientes reglas. Se puede resaltar que la segunda regla es recursiva:

$$ancestro(X, Y) :- padre(X, Y).$$

$$ancestro(X, Y) :- padre(X, Z), ancestro(Z, Y).$$

Al tiempo de añadir literales al lado derecho de cada regla, FOIL evalúa la utilidad de dichos literales. Para ello, aplica el concepto de *entropía* para estimar la cantidad de información de cada literal candidato, de igual forma que en ID3.

Para resumir, se puede mencionar que FOIL usa un lenguaje de representación poderoso, como es la lógica de predicados de primer orden y al mismo tiempo combina las principales ventajas de los algoritmos ID3 y de la familia AQ.

2.2.4 Redes Neuronales

Las redes neuronales (RRNN) surgieron como un modelo biológico de inteligencia, aunque hoy en día están consideradas como una forma de aprendizaje basada en la Teoría de la Psicología Conductista. Las RRNN tratan de modelar el cerebro humano. Freud (1930) en su libro “Teoría Sicodinámica” describe el cerebro como una red de neuronas [28]. El cerebro humano consta de 10^{11} neuronas conectadas una a otra por medio de sinapsis. Una sola neurona está conectada a otras neuronas por medio de unas 2000 sinapsis. Se dice que la inteligencia humana depende de la cantidad y calidad de las sinapsis entre las neuronas.

En 1962 Rosenblatt construyó el Perceptrón que hacía uso de las neuronas artificiales inventadas por McCulloch y Pitts en 1943. Un modelo básico de neurona es un elemento lógico con umbral que tiene n entradas (“sinapsis”) y una salida (“axón”). Cada entrada puede estar ponderada por un peso w_i comprendido entre -1 y $+1$; la salida es 1 si la suma de las entradas multiplicada cada una por su peso supera o iguala el umbral θ y 0 en caso contrario (ver Figura 3).

El estado de la neurona en cada momento está determinado por los valores de los pesos y del umbral. La información está codificada en los pesos w_i . Cada entrada representa un atributo del ejemplo que se quiere aprender o clasificar. A los atributos relevantes se les asigna más peso que a los menos relevantes. El mecanismo de aprendizaje del Perceptrón

es un algoritmo que modifica los pesos durante la fase de entrenamiento. Por ejemplo: un algoritmo muy sencillo para el caso de una salida binaria consiste en actuar solamente en caso de error del siguiente modo: si para una determinada entrada la salida es 0 cuando debería haber sido 1, se deben incrementar los pesos correspondientes a las conexiones de manera que se obtenga una salida de 1; y si la salida es 1 cuando debería haber sido 0, se deben decrementar los pesos. El objetivo consiste en que después de un determinado número de ejemplos en la fase de entrenamiento, el sistema debe ser capaz de clasificar ejemplos sin errores o con un porcentaje de errores aceptable. Es claro que, un Perceptrón sólo puede aprender categorías simples. Se usa para clasificaciones en hiperplanos en espacios en cualquier dimensión.

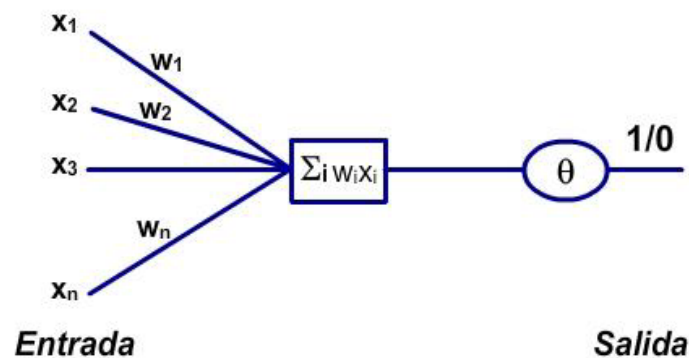


Figura 3. Ejemplo de un Perceptrón. Extraído de [28].

Minsky y Papert, en 1969, demostraron las limitaciones del Perceptrón, en especial la condición de que el sistema converja durante el entrenamiento cuando las categorías no son separables linealmente. Otros investigadores propusieron la introducción de capas de neuronas ocultas entre las neuronas de entrada y salida para mejorar el funcionamiento de la red.

Rumelhart, en 1986, desarrolló las Redes de Retropropagación en las que se incluían los niveles ocultos de neuronas y el algoritmo de refuerzo sobre los pesos intermedios [29]. Así, una red de retropropagación consta de: nodos de entrada, nodos de salida y nodos ocultos en niveles intermedios. En estado inicial, la red tiene pesos aleatorios. Se dispone de un conjunto de datos de entrada. Para cada conjunto de datos se generan valores de

salida que se comparan con las salidas deseadas. Si existe diferencia, se ajustan los pesos de las sinapsis. Se repite este proceso hasta que las respuestas sean exactas (dentro de un margen). Una vez estabilizada la arquitectura de la red, se la usa para clasificar nuevas entradas.

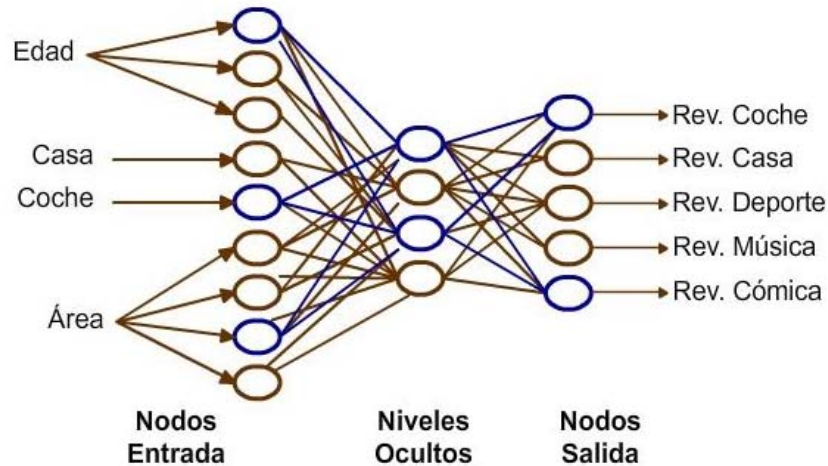


Figura 4. Ejemplo de un Perceptrón. Elaboración propia.

Se puede mostrar un ejemplo de datos de lectores de unas revistas de coches y cómicas. Los datos de estos lectores incluyen su edad, si tienen casa o no, si tienen coche o no y el código de área donde viven. En principio, los nodos de entrada están completamente interconectados con todos los nodos ocultos y éstos, a su vez, están completamente interconectados con todos los nodos de salida. En una red sin entrenamiento, todas las conexiones tienen el mismo peso. Durante la fase de entrenamiento, la red recibe ejemplos de entrada y produce salidas correspondientes a los datos de los lectores de las diferentes revistas, como se puede apreciar en la Figura 4. También se puede apreciar que la configuración interna muestra que existe cierta relación entre los lectores de revistas de coches y revistas cómicas, puesto que los atributos que más peso tienen son: edad, tener coche o no y el código de área. Sin embargo, la red neuronal no explica esta conexión.

Por lo tanto, las redes neuronales sirven para clasificar datos, incluso con ruido, aunque su entrenamiento puede ser lento y largo. Su desventaja principal es que el conocimiento generado es difícil de interpretar.

2.2.5 Aprendizaje No-Supervisado

El Aprendizaje No-Supervisado consiste en generar clasificaciones conceptuales a partir de objetos no clasificados [5] [22]. Se conforman grupos (clusters) de objetos que son similares entre sí usando alguna medida de similitud. La más simple de estas medidas es la distancia numérica entre objetos. Por ejemplo, en el gráfico mostrado en la Figura 5 se muestra la cantidad de revistas cómicas vendidas por una editorial según la edad. A primera vista se pueden distinguir dos grupos de personas: las personas con edad comprendida entre los 60 y 80 años y las personas con edad comprendida entre 10 y 35 años que compran revistas cómicas.

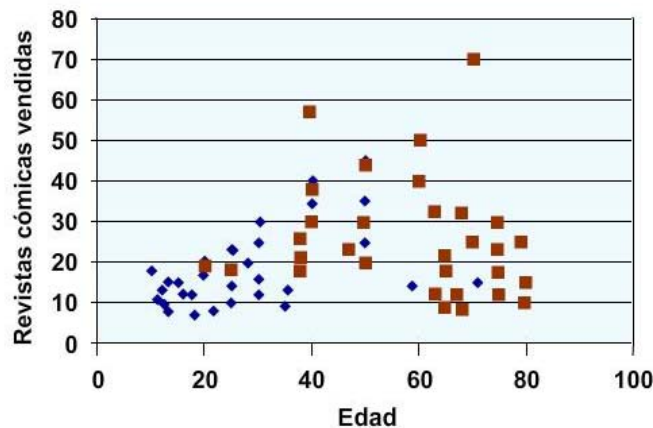


Figura 5. Cantidad de revistas cómicas vendidas según edad. Elaboración propia

De manera general, se pueden concebir los datos de los objetos a clasificar como puntos en el espacio k-dimensional, por lo tanto, se puede calcular la distancia entre los objetos. Los objetos que están cerca uno del otro son muy “parecidos”. Los objetos que están lejanos uno del otro representan a individuos que tienen poco en común [39]. Se puede medir la distancia entre dos puntos usando la distancia Euclídea, Manhattan, etc.

La distancia Euclídea entre dos registros con los campos: edad, ingreso y crédito se calcula como la raíz cuadrada de la suma de la diferencia al cuadrado de sus componentes (ver Tabla 2).

	Edad	Ingreso (miles euros)	Crédito (miles euros)
Cliente 1	32	40	10
Cliente 2	24	30	2
Diferencia	8	10	8
$\sqrt{\text{Suma cuadrados}}$	$(8^2 + 10^2 + 8^2) = 15$		

Tabla 2. Datos ficticios de clientes

La distancia entre el cliente 1 y el cliente 2 es de 15. Es importante que todos los atributos tengan una escala del mismo orden de magnitud para obtener medidas de distancias confiables entre registros. De esta forma, los registros pueden ser vistos como puntos en el espacio multidimensional de los datos.

Se puede definir el concepto de *vecindad* como el conjunto de registros que están uno cerca de otro (o son *vecinos*). El concepto de *vecino más próximo* sirve para predecir el comportamiento de un individuo que es *vecino* en una *vecindad*. Es de suponer que los individuos de la misma *vecindad* deberían tener comportamiento parecido. Así, si se desea predecir el comportamiento de un individuo se deben buscar los *k-vecinos* más cercanos a él, se calcula el promedio de comportamiento de estos *k-vecinos* y éste promedio es la predicción del individuo.

El concepto de *vecino más próximo* es sencillo y al mismo tiempo es poderoso, puesto que sirve como procedimiento inicial para tener un estimado de la agrupación de los individuos. Es apropiado para muestras de tamaño limitado, pero al tener complejidad $O(n^2)$, puede tomar mucho tiempo, si se disponen de muchos datos. Además, no trabaja bien en *k*-dimensiones, cuando *k* es muy grande, puesto que los puntos en el espacio se dispersan demasiado y no se distinguen las agrupaciones.

Una *vecindad* o *cluster* se entiende como un grupo de objetos que juntos representan un concepto. Para determinar *k-vecindades* en un conjunto de objetos es conveniente conocer el número de *vecindades* a priori. Luego se procede a elegir *k* semillas y se las trata como si fueran centroides de las *k* diferentes *vecindades*. Se distribuyen el resto de

los registros asignándoles a la *vecindad* cuya distancia con el centroide sea menor. A continuación, se recalculan los centroides como media aritmética o moda de los elementos de la *vecindad*. Se comprueba si el algoritmo ha alcanzado una posición estable, esto es, si los centroides actuales son iguales a los centroides de la iteración anterior, entonces se han determinado las *k-vecindades*; caso contrario, se repite el procedimiento.

Este algoritmo es muy popular, pero muy sensible al parámetro *k*: un valor de *k* superior al valor real dará lugar a clases ficticias y un valor de *k* inferior al valor real producirá menos clases de las existentes.

2.2.6 Aprendizaje por Refuerzo

Uno de los enfoques más usados dentro del Aprendizaje Automático es aquel que a partir de ejemplos (pares entradas – salidas provistos por el ambiente) predice la salida de nuevas entradas. Para reforzar o recompensar las acciones correctas es importante conocer la secuencia de acciones entre la entrada y la salida.

Dentro del Aprendizaje por Refuerzo los conceptos clave son: estado y acción. Los estados definen las posibles situaciones del sistema y las acciones son aquellas funciones que el sistema puede realizar en un momento (estado) determinado. El objetivo del Aprendizaje por Refuerzo es aprender cuál es la mejor acción a ejecutar, cuando el sistema se encuentra en cierto estado [4] [28].

Existen varias formas de Aprendizaje por Refuerzo. En la *recompensa retrasada* el proceso de aprendizaje es así: Inicialmente se ejecutan acciones de forma aleatoria desde cualquier estado. Cuando una de esas acciones produce el resultado deseado, entonces, se recompensa. El problema es que no se conoce el resultado inmediatamente después de una acción, sino después de una larga secuencia de acciones. Y las recompensas suelen tomar sólo dos valores: 1 ó 0 (correcto o incorrecto).

También se puede mencionar al aprendizaje con *recompensa acumulada* que es aquel obtenido durante todas las acciones ejecutadas por el sistema. La recompensa es tanto mayor cuánto antes se alcance el resultado deseado (por ejemplo: menos acciones realizadas).

Finalmente, el aprendizaje Q tiene como objetivo inferir la función $Q(s,a)$, donde s es el estado y a es la acción a realizar. $Q(s,a)$ representa la máxima recompensa que se podría obtener desde el estado s si:

- a) La primera acción que se realiza es la acción a
- b) Las acciones posteriores se suponen perfectas (las mejores posibles)

Conocido $Q(s,a)$, el sistema sabe qué acción tiene que ejecutar en cada estado: la acción a elegir es la que ofrece el valor máximo para $Q(s,a)$. El valor de Q se consigue así: inicialmente $Q(s,a)$ toma valores aleatorios (a cada estado y a cada acción posible en ese estado se les asigna un valor Q aleatorio). Comenzando del estado inicial, se realizan acciones aleatoriamente hasta que se cumpla alguna de las siguientes condiciones: el sistema alcance el objetivo deseado o el número de acciones realizadas alcance el límite máximo. Luego, se actualizan los valores de Q en función del resultado obtenido: si se ha alcanzado el objetivo, se recompensan las acciones proporcionalmente a la distancia al objetivo final o si se ha realizado el número límite de acciones sin alcanzar el objetivo final, no hay recompensa. Esta secuencia de pasos se repite hasta que Q se aproxime a un valor preestablecido de precisión o se alcance un número máximo de repeticiones [4].

Se ha demostrado que es posible obtener Q con total precisión si se recorren todos los estados y todas las acciones un número infinito de veces. En la práctica no es necesario alcanzar una precisión total, pero aún así se requiere un número de iteraciones muy elevado.

2.2.7 Razonamiento Basado en Casos

La idea principal del Razonamiento Basado en Casos (RBC) es resolver un nuevo problema recuperando situaciones previas similares y reutilizando la información y el conocimiento de dichas situaciones [31]. En el RBC, el conocimiento está especificado en una base de casos. Cada caso contiene la descripción del problema y su solución o estrategia de solución.

Para resolver un problema se empareja el caso actual con los casos existentes en la base de casos con el propósito de recuperar los casos más parecidos. Si dos casos se parecen, se supone que sus soluciones también los serán. Por lo tanto, los casos recuperados se usan para sugerir soluciones al problema actual. Si es necesario, la solución propuesta se puede revisar o evaluar para adecuarla al caso actual. Finalmente, cuando el caso actual está resuelto, se lo almacena en la base de casos, junto con su solución final.

El RBC sigue el siguiente proceso:

- Recuperar el caso, o casos, más parecido al caso actual de la base de casos.
- Reusar el caso recuperado para tratar de encontrar solución al problema actual
- Revisar y adaptar la solución propuesta, si es necesario
- Almacenar la solución y problema actual en la base de casos

Existe una variedad muy grande de técnicas para organizar, recuperar e indexar el conocimiento. Para recuperar un caso, por ejemplo, se debe identificar el conjunto relevante de descriptores del problema, emparejar el caso y devolver un conjunto suficiente de casos similares y finalmente elegir el mejor caso del conjunto de casos seleccionado. El re-uso de la solución del caso recuperado en el caso actual incluye la identificación de las diferencias entre ambos y el reconocimiento de la parte que puede ser transferida al caso actual. La revisión de la solución generada para el caso actual se realiza cuando dicha solución fue incorrecta. Se considera una forma de aprendizaje. Finalmente, el almacenamiento del nuevo caso es útil para alimentar la base de casos.

Este proceso incluye la decisión de qué información guardar, en qué formato almacenarla, cómo indexarla y cómo integrarla a la base de casos [49].

El RBC apareció en herramientas comerciales a principios de los 90 y se han creado aplicaciones en diversos dominios tales como el diagnóstico médico, asistencia al usuario, toma de decisiones, diseño, asesoramiento en áreas de finanzas, etc. En el Departamento de Ingeniería de Software e Inteligencia Artificial, específicamente en el Grupo de Aplicaciones de Inteligencia Artificial de la Facultad de Informática de la Universidad Complutense de Madrid se ha desarrollado una herramienta para construir sistemas de Razonamiento Basado en Casos llamado JColibri [38].

3. Comercio Electrónico Basado en Agentes

3.1 Introducción

El comercio electrónico plantea un nuevo marco de negocios para desarrollar operaciones mercantiles. En España, según [19], el comercio electrónico ha alcanzado la cifra de 126.909 millones euros en el año 2007 y crece a un ritmo del 32.7% interanualmente. Se puede definir al comercio electrónico “como cualquier forma de transacción comercial donde las partes interesadas interactúan electrónicamente, en lugar de por intercambio o contacto físico directo” [46]. Dentro los procesos de comercio de las empresas se incluyen las tareas de mercadotecnia, de contabilidad, auditoría en línea, banca electrónica, tele-trabajo, sistemas de información, tiendas virtuales, subastas electrónicas, etc. Por supuesto, los procesos de compra y venta de productos son los aspectos más exitosos del comercio electrónico, en sus diferentes modalidades:

- B2C (Business to Consumer). Consiste en la venta de productos finales a un consumidor, por ejemplo, las tiendas virtuales como: <http://www.amazon.com>
- B2B (Business to Business). Comercio entre empresas, generalmente una empresa cliente y otra empresa proveedor, por ejemplo: <http://www.opcion.com>
- C2C (Consumer to Consumer). Subastas en las que usuarios particulares venden productos a otros. Se puede citar a: <http://www.ebay.com>
- C2B (Consumer to Business). Los consumidores particulares se agrupan para tener más fuerza y hacer pedidos a empresas. Por ejemplo, cooperativas como: <http://www.letsbuyit.com>
- A2B (Administration to Business). Relaciones con las administraciones públicas y los ciudadanos, empresas u otras administraciones. Por ejemplo, impuestos vía Internet: <http://www.aeat.es>
- B2E (Business to Employee). Comunicación entre empresa y trabajador. Por ejemplo: reclutamiento electrónico, tele-trabajo, etc.

- P2P (Peer to Peer). Intercambio de amigo a amigo, por ejemplo de música o los préstamos personales como: <http://www.prosper.com>

El B2C es la modalidad de comercio electrónico más conocida por el público y es la modalidad que se va a utilizar en este proyecto. Trata de la venta de productos finales al consumidor a través de Internet, en las denominadas tiendas virtuales. Las tiendas virtuales disponen de un catálogo de productos donde el cliente examina una lista de productos a ser vendidos, elige el producto que más le conviene y finalmente paga por éste usando algún medio de pago (por ejemplo: tarjeta de crédito, transferencia bancaria, etc.).

3.2 Infraestructura para el Comercio Electrónico

Según Carles Sierra, para implantar el comercio electrónico donde interactúen agentes software se deben disponer de tres elementos fundamentales: organización, mecanismos de interacción y confianza [47].

a) Organización: La primera consiste en una estructura que brinde los parámetros que regulen el ambiente para los agentes. Como los agentes pueden ser autónomos, heterogéneos, independientes, no-benevolentes, no-fiables y responsables, se tienen problemas con la coordinación y la cooperación. Por lo tanto, se hace necesaria una tercera parte que establezca y normalice las interacciones, asigne riesgos, garantice y salvaguarde ciertas acciones y prevenga situaciones no deseadas. Esta es la noción intuitiva de institución.

Así, “una institución tiene como objetivo principal facilitar, supervisar y hacer cumplir los compromisos entre los participantes en situaciones repetitivas” [47]. Una institución electrónica debería tener los siguientes tres componentes:

- i. **Estructura de diálogo:** Se deben contemplar los aspectos ontológicos de la institución, tales como el lenguaje a comunicar, las primitivas de

comunicación, su significado y las entidades involucradas en la comunicación.

- ii. **Estructura ejecutiva:** Incluyen las convenciones para establecer el flujo de interacciones y las consecuencias sociales de las acciones en la institución.
- iii. **Normas para el comportamiento individual:** Son las convenciones que los individuos siguen para actuar dentro de la institución.

b) Mecanismos: Los mecanismos establecen las reglas para convenir la interacción entre los agentes, de manera que exista equilibrio entre ellos. Si se definen las reglas de juego, se debe enfocar cómo deben realizarse las interacciones, cómo manejar las restricciones en la autonomía y determinar ciertos tipos de comportamiento racional. Por ejemplo: uno de los mecanismos más importantes para la interacción de los agentes es evitar la revelación de las preferencias de los agentes.

Cuando se construye un agente software con capacidad de negociación se debe considerar lo siguiente:

- i. El protocolo de negociación a usar
- ii. Los objetos que se negocian
- iii. El modelo de razonamiento que el agente emplea

c) Confianza: Si existe confianza entre los agentes, se reducen los riesgos de interacción. También, para reducir los riesgos de agentes en comercio electrónico, los canales de comunicación entre los participantes deben incluir capacidades de confidencialidad, integridad, autenticación y no-repudio. Los sistemas distribuidos, formados por miles o millones de agentes necesitan de mecanismos de seguridad, sobre todo en aplicaciones de comercio electrónico, donde se maneja dinero.

3.3 Etapas del Comercio Electrónico

El comercio incluye todas las interacciones entre compradores y vendedores en todas sus etapas: búsqueda, compra y entrega. En el modelo de Comportamiento de Compra del Consumidor algunos autores como [16] y [20] han identificado seis etapas:

- a) Identificación de la necesidad de compra
- b) Elección del producto a comprar
- c) Elección del vendedor
- d) Negociación de las condiciones de compra
- e) Realización de la compra y entrega del producto
- f) Uso y evaluación del producto del producto adquirido

Muchos de estos pasos continúan sin automatizarse [13], tal es así que muchas personas realizan la búsqueda, compra y pago de productos en la Web por sí mismas. Sin embargo, actualmente existen formas innovadoras para implementar el B2C con el uso de agentes software. Pattie Maes [25], fundadora del grupo de investigación “Software Agents” del Massachusetts Institute of Technology (MIT), concibe a un agente software como un programa informático que actúa como un asistente del usuario aprendiendo paulatinamente de la interacción con el usuario, y que con el transcurso del tiempo, es capaz de anticiparse a sus necesidades [27]. De esta forma, se pueden pensar en agentes software con capacidades inteligentes que puedan operar en los mercados electrónicos. En la figura 6 se puede apreciar la relación existente entre las funcionalidades inteligentes que pueden tener los agentes, las etapas de compra y los posibles efectos en el mercado [20].

Los agentes buscadores tratan de satisfacer las necesidades de los compradores informando de los productos que puedan interesarle, de acuerdo a su historial de compra [2]. Por ejemplo, la tienda virtual <http://www.amazon.com> dispone de un agente que anuncia ofertas de libros y los nuevos libros disponibles. Incluso puede notificar al usuario del resultado de su búsqueda a través de correo electrónico.

Los agentes de recomendación actúan como representantes de los vendedores, haciendo predicciones de los gustos de los clientes, tomando en cuenta su perfil. Se puede nombrar al destacado Firefly [26] que usa la técnica de filtrado colaborativo para recomendar productos. Los autores mencionan que el portal Yahoo! utiliza Firefly para personalizar sus listas “top” de música y películas.

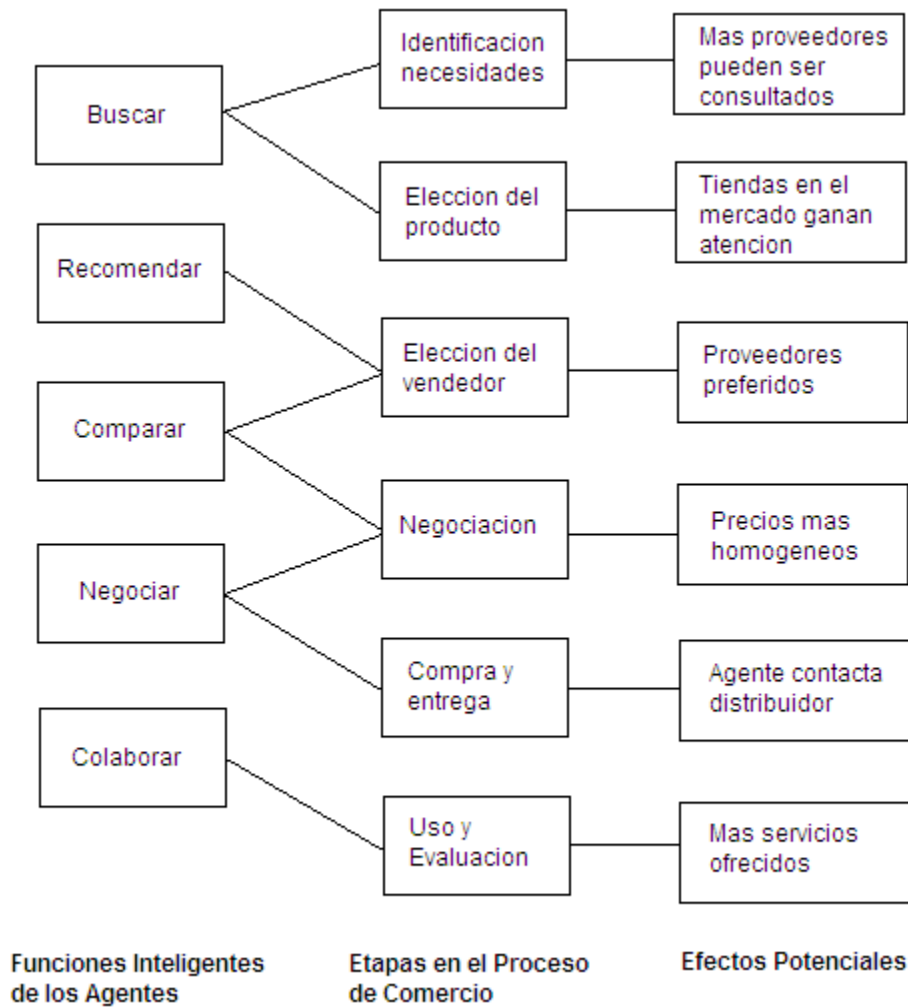


Figura 6. Funciones de los agentes inteligentes en el comercio electrónico. Adaptado de [20]

Asimismo, los agentes de comparación extraen información de múltiples sitios acerca de los productos que desea el usuario. Según un estudio de Dataquest [46], en la actualidad,

un 10% de las ventas en Internet se canalizan por este medio. Se puede mencionar a Bargain Finder [21], como al primer agente comparador exitoso, que busca en varias tiendas virtuales de música los discos compactos con los precios más baratos. Las compras se limitan a los comerciantes que se suscriben y los clientes tienen que saber deletrear correctamente los nombres de los artistas y álbumes.

Este tipo de agentes realizan comparaciones en línea de productos de manera más eficiente que las que hacen los humanos convencionalmente. De esta forma, favorecen la competencia entre las empresas productoras, al mismo tiempo que reducen el tiempo y costo de búsqueda por parte de los compradores. Algunos autores sostienen que usar agentes para buscar productos en Internet tiene un “costo marginal igual a cero” [9].

Los agentes mencionados previamente facilitan la conexión entre potenciales compradores y vendedores. Esto representa las tres primeras etapas del comercio. Para completar la transacción los agentes deberían negociar un trato aceptable para ambos y luego realizar el intercambio de bienes. Una aproximación a ello es el sistema Kasbah [7] que automatiza el proceso de compra y venta de productos a través de agentes compradores y vendedores que usan diferentes estrategias de negocio para conseguir el “mejor trato posible”. También se puede mencionar a MAGMA [48] que ofrece un mercado virtual más completo puesto que incluye un banco, infraestructura de comunicación, publicidad y agentes que negocian usando el protocolo de Vickrey.

3.4 Proceso de Negociación

El proceso de negociación es un componente clave del comercio electrónico [44]. En la negociación automática, los agentes computacionales tienen que ponerse de acuerdo, en lugar de las personas que representan. Esta automatización ahorra tiempo y situaciones embarazosas a las personas y muchas veces los agentes computacionales consiguen mejores tratos en situaciones donde existen complejas combinaciones de variables. En general, la negociación no solamente comprende el regateo sobre el precio del producto, sino también en satisfacer las diferentes preferencias de los usuarios.

La negociación automática se puede aplicar a diferentes áreas, como son el comercio electrónico, los mercados de electricidad, la asignación de banda ancha, la planificación de manufactura, la programación de redes subcontratadas, el despacho distribuido de vehículos, etc.

3.4.1 Definición de Negociación

El concepto de negociación tiene varias definiciones. Entre ellas, se puede mencionar la que señala Rahwan et al en [37]: “Negociación es una forma de interacción en la cual un grupo de agentes, con intereses conflictivos y deseos de cooperar, tratan de alcanzar un acuerdo mutuamente aceptable acerca de la división de los recursos escasos”. Al término del proceso, todas las partes deben estar convencidas de que tuvieron oportunidades de argumentar, de que sus ideas fueron tomadas en cuenta y plenamente entendidas y que el resultado final fue la suma de sus contribuciones individuales [11].

El proceso de negociación de un producto puede darse por diferentes criterios, o por varios a la vez: el precio, la cantidad de artículos, el plazo de entrega, las condiciones de entrega y pago, disponibilidad, etc. En general, el regateo se relaciona con la negociación por precio y en este trabajo nos abocaremos a este tema.

3.4.2 Zona de Acuerdo

En una negociación con posibilidades de éxito, las funciones de preferencia de las partes involucradas están negativamente correlacionadas. Es decir, un comprador paga un determinado valor por un producto porque entiende que el producto vale más de lo que pagará por éste. El vendedor, a su vez, vende el producto porque entiende que éste vale menos del valor que se pagará por él. De esta manera, los intereses del comprador y del vendedor son inversos y es factible negociar.

El concepto de “zona de acuerdo” permite entender el concepto de negociación o regateo (ver Figura 7). Se puede pensar que un comprador y un vendedor están negociando el precio de un producto o servicio y cada una de estas partes tiene valores límite para la negociación. El vendedor tiene su precio de reserva V que es el precio mínimo en el que puede vender el producto. Para cualquier valor de acuerdo final, X , el vendedor obtiene lucro si ocurre $X > V$. Obviamente, el vendedor desea maximizar su ganancia, es decir vender más alto que V . El precio de reserva del comprador es C y cualquier valor $X < C$ permite que el comprador, también, obtenga lucro.

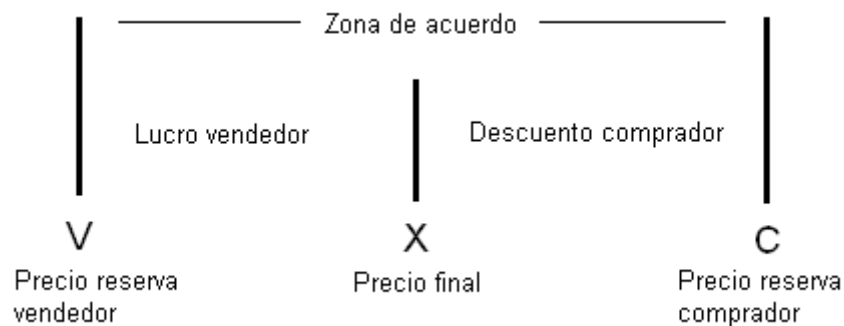


Figura 7. Zona de acuerdo en negociación. Adaptado de [11]

Si el precio de reserva V del vendedor es menor que el precio de reserva C del comprador, entonces existe una zona de acuerdo donde se puede negociar y el precio final X puede definirse a través de algún procedimiento de regateo. Si existe una zona de acuerdo, cada una de las partes deberá realizar concesiones a partir de su propuesta inicial. Durante la negociación, el vendedor tratará que X se mueva a la derecha y el comprador a la izquierda. A medida que ambos negocian en dirección a un acuerdo, se va definiendo el valor de X . El acuerdo se establecerá después de varias consideraciones hasta culminar en un punto X que determina el acuerdo final.

Es verdad que, el comercio electrónico favorece a que se encuentre fácilmente información sobre los productos y sus precios, permitiendo que el comprador pueda localizar al proveedor que ofrezca el mejor precio y se reduzca el margen de negociación. Sin embargo, los estudios han demostrado que existe una variación significativa en los

precios de los mismos productos en diferentes tiendas virtuales, dando una mayor posibilidad de negociación [11]. Otro punto a favor de la negociación es que a los compradores les entusiasma regatear y no sólo por la parte económica, sino porque ven al proceso de negociación como un juego de desafío, donde pueden participar de manera entusiasta.

3.4.3 Estructura para Negociación Electrónica

Rahwan et al cita en [37] la propuesta de Bartolini et al acerca de una estructura para caracterizar los componentes de una negociación electrónica (ver Figura 8). Los componentes son:

- a) **Ambiente de negociación:** La plataforma de comunicación a través de la cual los agentes interactúan.



Figura 8. Estructura de Negociación. Adaptado de [37]

- b) **Monitor:** El que gestiona el ambiente de negociación y facilita la negociación. Se asegura que los participantes cumplan el protocolo de negociación, paguen las multas si violan las reglas, etc.

- c) **Plantilla de negociación:** El lenguaje de negociación para describir los acuerdos.
- d) **Reglas de negociación:** Un conjunto de reglas que incluyen:
- Reglas para admitir a los participantes
 - Reglas para validar las propuestas
 - Reglas para hacer propuestas
 - Reglas para hacer peticiones
 - Reglas para actualizar información de los participantes
 - Reglas para hacer conocer el estado de la negociación
 - Reglas para especificar si se ha llegado a un acuerdo final
 - Reglas para terminar una participación
 - Reglas para terminar la negociación
- e) **Número de participantes:** Incluye para cada participante la definición de su comportamiento interno y su estrategia de negociación.

3.4.4 Desempeño de la Negociación

Las medidas económicas para medir el desempeño de un proceso de negociación representan un punto de referencia con respecto al desempeño de una negociación óptima en términos de eficiencia e imparcialidad. Las medidas típicas usadas en diferentes estudios de negociación [15] comprenden:

- **Conjunto de beneficios/utilidad.** Se define como la suma total de utilidades, incluyendo las ponderaciones en las variables que la componen, tanto del comprador como del vendedor, cuando se llega al acuerdo final. Lo más importante para un negociador es conseguir la utilidad más alta posible. Las utilidades parciales se usan para calcular la utilidad total.
- **Balance de contrato.** Se calcula como el valor absoluto de la diferencia entre el total de la utilidad conseguida por cada negociador. Si el resultado es cero, entonces el acuerdo final está balanceado, si el resultado es alto, entonces el acuerdo no está balanceado.

- **Distancia a la frontera eficiente.** Mide la eficiencia del acuerdo final. Todas las soluciones eficientes en la frontera eficiente deben ser calculadas antes de calcular el valor de la distancia D_1 .

$$D_1 = \min_{i=1}^n \sqrt{(F_c - E_{ci})^2 + (F_v - E_{vi})^2} \quad (\text{Ecuación 1})$$

F_c y F_v denotan la utilidad final del comprador y del vendedor respectivamente
 E_{ci} y E_{vi} denotan la solución eficiente i del comprador y del vendedor, respectivamente.

Se usa i como índice secuencial de las soluciones eficientes y n es el número total de soluciones eficientes.

- **Distancia a la solución de regateo Nash.** Mide la imparcialidad del acuerdo final. La solución de regateo Nash se establece como el valor que maximiza las utilidades de los dos negociadores. La solución Nash debe ser determinada antes que pueda ser calculada la distancia D_2 .

$$D_2 = \sqrt{(F_c - N_c)^2 + (F_v - N_v)^2} \quad (\text{Ecuación 2})$$

Donde F_c y F_v denotan la utilidad final del comprador y del vendedor, respectivamente.

N_c y N_v denotan la solución final de regateo Nash, respectivamente.

3.4.5 Sistemas de Información de Apoyo a los Procesos de Negociación

El aspecto teórico intenta predecir el proceso y el resultado de la negociación, pero en la práctica, se trata de ayudar a las personas a negociar efectivamente. Las investigaciones muestran que las personas, cuando negocian, aceptan la primera propuesta que les satisface. Sin embargo, frecuentemente sienten que “podían haber dejado menos dinero del que dieron” si hubiesen continuado el juego [15]. De esta forma, los sistemas de

información pueden ayudar de manera determinante a mejorar el proceso de negociación y sus resultados asistiendo a los humanos o negociando por ellos.

Los sistemas de información para apoyar a los procesos de negociación pueden ser:

- **Mensajes electrónicos.** Se pueden usar diferentes herramientas para apoyar la negociación en línea, tales como: correo electrónico y los sistemas de mensajes instantáneos.
- **Sistema de soporte de negocios.** Son sistemas que incorporan herramientas de toma de decisiones para asistir a las partes negociadoras a alcanzar un acuerdo, ofreciendo apoyo para la resolución de problemas analíticos complejos. También pueden incluir algún sistema de mensajes electrónicos.
- **Agentes software de negociación.** Los estudios recientes de agentes software inteligentes que se detallaron en el apartado 3.3 han abierto un mundo de posibilidades para automatizar la negociación en el comercio electrónico, permitiendo que los agentes software puedan negociar por los humanos. Los agentes software de negociación permiten encontrar mejores tratos en ambientes complejos y ahorran tiempo a los humanos.

Se puede comprender fácilmente que los negociadores que usan sistemas de soporte de negocios tienen mejores resultados que los que negocian sólo a través de mensajes electrónicos y que una negociación sin apoyo computacional es menos eficiente que una que por lo menos usa mensajes electrónicos con este fin. En general, un sistema de soporte de negociación apoyado con mensajes electrónicos tiene más éxito, que funcionando de manera aislada. Finalmente, los agentes software de negociación actuales pueden negociar de forma comparable a un humano, pero aún les faltan capacidades de aprendizaje para tener mejor desempeño.

4. Desarrollo del Sistema de Negociación Automática

4.1 Introducción

De acuerdo a la Asociación Nacional de Gestión de Compras (en inglés, NAPM: National Association of Purchasing Management) el promedio para hacer una compra importante es de 12 semanas y la mayor cantidad de tiempo y esfuerzo se emplea en los procesos de negociación y de resolución de conflictos [36]. Por ello, la importancia de automatizar el proceso de negociación.

Actualmente, las técnicas de Aprendizaje Automático están siendo ampliamente usadas en los sistemas de negociación para llevar a cabo negociaciones efectivas e inteligentes y en representación de los clientes. También se ha visto que para conseguir mejor desempeño en el proceso de aprendizaje es necesario disponer de información de negociaciones pasadas.

4.2 Trabajos Relacionados

En la literatura investigada se ha podido apreciar que existen trabajos relacionados con la Automatización de la Negociación, pero desde diferentes puntos de vista. Algunos autores visionarios como Maes [27][26] señalan que a los agentes software se les puede delegar las diferentes etapas del comercio electrónico (ver 3.3), incluyendo el proceso de negociación, de tal manera que ahorren tiempo y dinero a los usuarios y describen algunos ejemplos de agentes que tienen un comportamiento interesante, pero no inteligente. En la misma dirección, Sierra [47] y He et al [17] discuten de manera teórica acerca de los desafíos y oportunidades que brinda el comercio electrónico, y específicamente la negociación, basada en agentes. Algunos autores plantean un enfoque sistemático acerca de la negociación, como Lomuscio et al [24] que determinan los principales parámetros que influyen en la negociación automática. Rahwan et al en [37] proponen la metodología STRATUM para diseñar las estrategias de negociación de los

agentes en diferentes entornos. Otros autores como [11] y [15] hacen estudios y experimentos del proceso de negociación en comercio electrónico y plantean algunas estrategias para conseguir mejores resultados en la negociación de productos.

En la parte práctica, existen muchos trabajos que han desarrollado agentes software que pueden participar en subastas electrónicas. Por ejemplo, ITA (Intelligent Trading Agency) de [36] propone una infraestructura uno-a-muchos para negociación concurrente en subastas. Existe un agente por subasta que negocia por el usuario usando técnicas de razonamiento basadas en restricciones. También se puede mencionar al famoso AuctionBot [16] que es un agente que puede participar en diferentes tipos de subastas (por ejemplo: inglesa, de Vickrey, etc.) pujando en lugar del usuario. Su razonamiento es puramente matemático. Chapman [6] propone un agente que puede aprender de su oponente, mientras participa en una subasta. Aplica teoría de probabilidades.

También se puede citar a Kasbah [7] que presenta un mercado virtual donde existen agentes vendedores y compradores que negocian entre sí para conseguir el “mejor precio posible”. Estos agentes negocian en nombre del usuario, pero usando un algoritmo de negociación basado en funciones matemáticas sencillas. MAGMA [48] es otra aproximación más completa de mercado virtual que incluye un banco, infraestructura de comunicación, publicidad y agentes negociadores. Los agentes negociadores tienen dinero y se aplica la estrategia de Vickrey para la adquisición de los productos. TRACONET (TRANsportation COoperation NET) [45] es un sistema basado en agentes que negocia la re-asignación de trabajos entre diferentes compañías de flete. Su aportación es que es un sistema distribuido para la negociación automática entre diferentes compañías. Sin embargo, en todos estos ejemplos previamente mencionados los agentes implantados no son inteligentes y representan, más bien, la primera generación de agentes con capacidad de negociación en mercados electrónicos.

En cuanto al uso de técnicas de aprendizaje automático Zhang [50] las adopta para preservar la privacidad de la negociación aplicando Q-Learning para conseguir una negociación segura. Asimismo, Brzostowski [1] usa razonamiento basado en casos para

seleccionar al mejor vendedor con el cual negociar, usando como referencia situaciones previas. También, se puede mencionar que Lau [23] ha creado agentes negociadores adaptativos para B2B aplicando algoritmos genéticos. Chen [8] usa algoritmos genéticos que combina con teoría de juegos para implementar un mecanismo automático de negociación. Ambos trabajos tratan de conseguir el equilibrio de Nash y el Pareto óptimo, después de una serie de evoluciones. Fatima et al [12] aplican teoría de juegos y algoritmos genéticos para que los agentes puedan negociar de manera óptima. El inconveniente de conseguir la solución óptima es que demanda muchos supuestos, información completa del adversario (que casi nunca se tiene) y muchos recursos computacionales. Huang [18] ha diseñado un agente vendedor que principalmente pueda persuadir al comprador a realizar la compra, de la misma manera que lo hacen los vendedores humanos. Utiliza técnicas de aprendizaje por refuerzo y redes neuronales, pero sólo incluye una técnica de negociación básica.

4.3 Propuesta de Negociación Automática

El objetivo principal de este proyecto es automatizar la negociación en mercados electrónicos usando agentes software inteligentes. Como lo comentamos anteriormente, esta idea está basada en los conceptos generados por Pattie Maes [25][26][27] que concibe a un agente como un asistente del usuario, más que sólo como una herramienta informática. El agente a desarrollar debería aprender de la interacción con el usuario cómo negociar y en el futuro, debería negociar por él.

En el apartado de Trabajos Relacionados (ver 4.2) se puede apreciar que hasta el momento no se han desarrollado agentes inteligentes que sean capaces de aprender a negociar por el usuario y con el tiempo sean capaces de llevar a cabo acuerdos exitosos por el usuario. Existen muchos trabajos relacionados con el desarrollo de agentes software que pueden participar en subastas electrónicas [6] [18]. Las técnicas que usan para calcular las pujas están basadas principalmente en teorías estadísticas. Estos agentes consiguen buenos tratos, pero su desventaja es que no se comportan como asistentes personales, sino son sólo programas que calculan resultados. Otros autores han

desarrollado algoritmos de negociación para conseguir el precio óptimo [12]. Dichos algoritmos se basan en teorías de juegos, algoritmos evolutivos o teorías económicas y su objetivo es conseguir el valor máximo global en una negociación. Sus resultados son teóricos y no aplicables directamente a aplicaciones en línea de comercio electrónico, puesto que no es posible disponer de todos los datos del adversario y el cálculo del valor óptimo puede tomar varios minutos o incluso horas. Además, muchas veces el usuario se conforma con un valor óptimo local que se consiga rápidamente, que con uno global que consume muchos recursos. Finalmente, muchos autores hacen hincapié en la arquitectura interna de los agentes y en su organización, con un enfoque más bien metódico que práctico [30][37][47].

Entonces, basados en los estudios y análisis mencionados, principalmente se usó el trabajo de Chávez y Maes [7] para construir el primer prototipo de un Sistema de Negociación Automática, con agentes que tuvieran capacidad de negociar el precio de un producto por sus clientes. Dichos agentes, de la misma forma que en [7] usan funciones matemáticas para calcular el precio a ofertar, siguiendo un protocolo de negociación, pero sin ser inteligentes. A continuación, y después de haber validado este primer prototipo de Sistema de Negociación, se incorporó capacidad de aprendizaje a los agentes negociadores, para que aprendieran de sus usuarios, con cada interacción, a negociar mejor. En los siguientes apartados se describirán estas dos versiones de prototipo de Sistema de Negociación.

4.4 Prototipo de Sistema de Negociación Electrónica

En la Figura 9 se muestra el esquema del primer prototipo de Sistema de Negociación Electrónica (SNE) propuesto. En dicha Figura se puede apreciar a un agente Comprador, representante de un cliente humano y a un agente Vendedor, encargado de una tienda virtual. Ambos agentes pueden negociar el precio de un producto dentro un Mercado Electrónico. Los agentes interactúan usando un lenguaje simbólico de negociación, a través de una pizarra electrónica. El lenguaje de negociación incluye primitivas tales como: *ofertar, comprar, vender, trato hecho, salir*, etc.

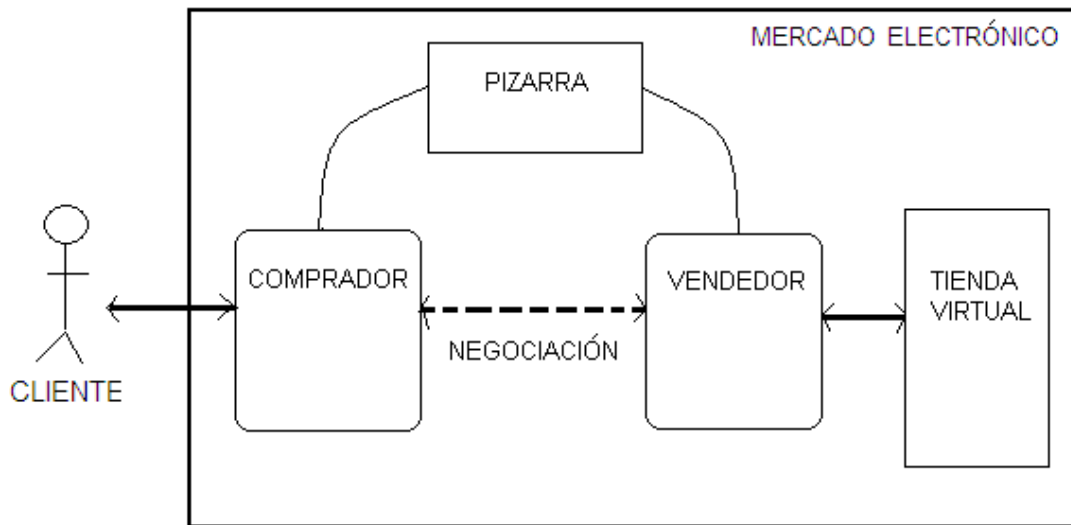


Figura 9. Arquitectura del Sistema de Negociación Electrónica (SNE)

La arquitectura de SNE incluye todos los componentes señalados en el apartado 3.4.3. El Mercado Electrónico se constituye en el local de negociación donde interactúan los agentes y en el monitor de su comunicación. Existe un lenguaje de negociación que permite que los agentes puedan ofertar, comprar, vender, ponerse de acuerdo, entrar al mercado, salir del mercado, etc. La comunicación entre los agentes sigue un protocolo de negociación para efectuar la compra y venta de productos. Finalmente, los participantes del mercado son: el agente Comprador y el agente Vendedor. En los siguientes acápite se describirán cada uno de estos componentes.

4.4.1 Agentes Software

Siguiendo la idea de Maes [27], se ha diseñado una plantilla de un agente negociador que pueda comportarse como un asistente personal del usuario. En este caso, el agente debe tener características que le permitan participar en un mercado virtual, ya sea en calidad de Comprador o de Vendedor. La plantilla creada se ha basado en el trabajo [7] y consta básicamente de los siguientes atributos:

- **Nombre:** Nombre del agente
- **Nombre del usuario:** Nombre del cliente o tienda al que representa
- **Producto:** Nombre del producto que se desea comprar o vender
- **Precio deseado:** Precio que se desea pagar o recibir por el producto
- **Precio máximo/mínimo a pagar:** Precio máximo a pagar o precio mínimo a recibir por el producto.
- **Precio inicial:** Precio inicial de negociación
- **Precio actual:** Cantidad ofertada en el momento actual
- **Precio actual adversario:** Cantidad ofertada por el adversario en el momento actual
- **Fecha inicial:** Fecha en la que se empieza el regateo
- **Fecha actual:** Fecha actual del proceso de regateo
- **Fecha límite:** Fecha tope en la que se debe realizar la compra o venta
- **Función incremento/decremento:** Función con variable acumuladora que permite calcular el nuevo precio a ofertar. Se tienen tres diferentes coeficientes para disponer de tres funciones que permitan calcular los precios con mayor o menor porcentaje de incremento (o decremento). Asimismo, cada función incluye un componente aleatorio para que el adversario no pueda deducir la estrategia de compra o venta del agente. La función tiene la siguiente forma:

$$p_i = (p_i + p_i * d_j) + a_i \quad i = 1..n, \quad j = 1..3 \quad (\text{Ecuación 3})$$

$$0 < a_i < (p_i + p_i * d_j) - p_i$$

donde p_i es el precio a calcular en la interacción i de la negociación; d_j es el porcentaje de incremento o de decremento; a_i es un valor aleatorio acotado entre el nuevo p_i calculado y p_i . Finalmente, n es el número total de interacciones en el proceso de negociación.

El precio ofertado se calcula así: el tiempo comprendido entre la *fecha inicial* y *fecha límite* se divide en tres partes: la primera parte se constituye en el primer 50% de tiempo, la segunda consiste en el siguiente 25% de tiempo y la última

parte, el 25% restante. En el primer periodo de tiempo de negociación se usa el coeficiente d_1 que representa el menor porcentaje de incremento o decremento (por ejemplo: 2.5% ó -2.5%); en el segundo periodo se usa el coeficiente d_2 que establece un porcentaje de incremento o decremento mayor (por ejemplo: 5% ó -5%) y en el tercero, se usa el coeficiente d_3 cuyo valor representa el mayor porcentaje de incremento o decremento de precio (por ejemplo: 7.5% ó -7.5%).

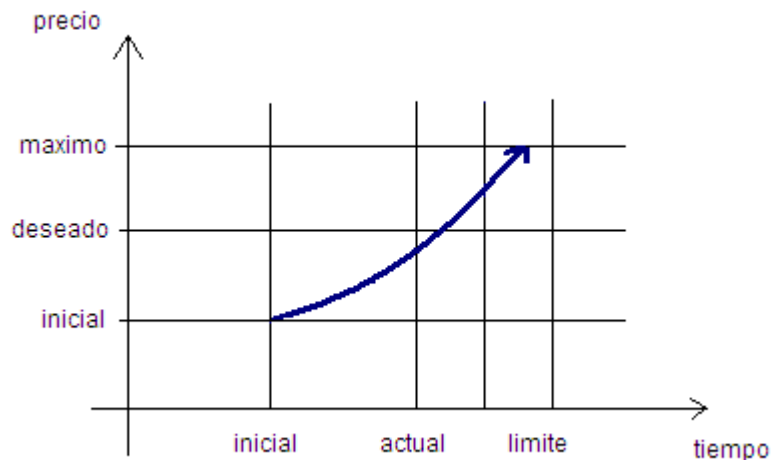


Figura 10. Representación de la función de incremento del Comprador con valores arbitrarios.

La razón es la siguiente: al principio no se tiene mucha prisa por comprar el *producto* y lo que se trata es conseguir el menor precio posible; entonces, se usa el coeficiente d_1 . En el segundo periodo de tiempo se usa una función matemática más agresiva con el coeficiente d_2 . En el último periodo se aplica d_3 , porque se tiene urgencia por comprar (vender) el *producto* y se puede ofrecer hasta el *precio máximo a pagar* (*precio mínimo a pagar*). Se puede ver en las Figuras 10 y 11 el comportamiento de las funciones de incremento y decremento de los agentes.

Se debe hacer notar que ni el agente Comprador ni el agente Vendedor conocen los valores correspondientes a los atributos de negociación de su adversario, esto es: *precio inicial*, *precio final*, *precio deseado*, *fecha inicial*, *fecha final*, *función incremento o decremento*. Algunos autores, como Fatima et al mencionan en [12] la influencia de

conocer la información del contrincante al momento de la negociación. Obviamente, si se dispone de mayor información, se puede aplicar alguna estrategia específica para conseguir un resultado exitoso, pero en nuestro planteamiento, el agente no dispone información de su contrincante, sólo la que se va conociendo a medida que se realiza el proceso de negociación.

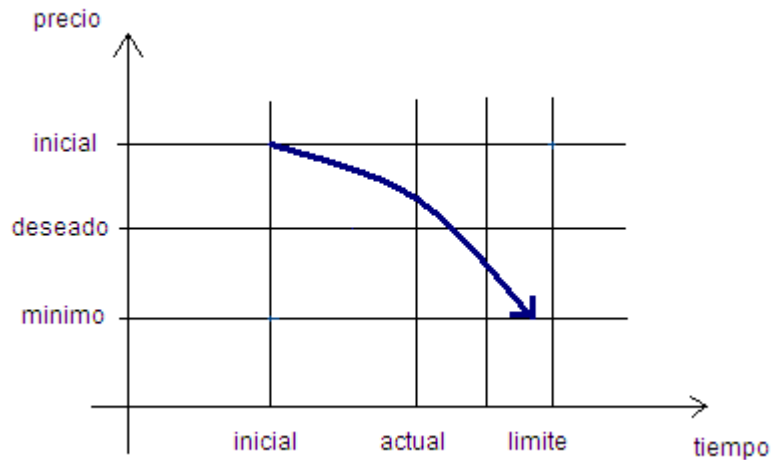


Figura 11. Representación de la función de decremento del Vendedor con valores arbitrarios

El comportamiento del Agente Comprador puede describirse así:

- **ElegirProducto:** Dado el catálogo del Vendedor, el agente Comprador puede elegir el *producto* buscado. Si existe el *producto* buscado, entonces el Cliente le proporciona los parámetros de compra, tales como: *precio inicial*, *precio deseado*, *precio máximo*, *fecha límite* de compra. Se supone que *fecha inicial* es la fecha en que ocurre este evento.
- **BuscarProducto:** El agente Comprador ingresa el nombre del *producto* buscado. Si existe, el Cliente le proporciona los datos iniciales de la compra: *precio inicial*, *precio deseado*, *precio máximo*, *fecha límite* de compra. La *fecha inicial* de negociación es la fecha en que ocurre este evento.
- **Ofertar:** El Comprador efectúa el proceso de regateo de precio mientras el precio ofertado por el Vendedor sea mayor a su *precio deseado* o *precio máximo* a pagar

y tenga tiempo para negociar. El precio ofertado se calcula usando la *función de incremento* explicada anteriormente.

- **Rechazar:** En cualquier momento se puede rechazar el *precio* propuesto por el agente Vendedor. Esta conducta sólo manifiesta una forma de disgusto ante la última oferta.
- **Comprar:** Si el *precio* del Vendedor es menor o igual al *precio* ofertado por el Comprador, entonces se compra el *producto*.
- **Salir:** En cualquier momento, el agente Comprador puede retirarse del proceso de negociación y abandonar el Mercado Electrónico.

Por su parte, el agente Vendedor tiene el siguiente comportamiento:

- **Presentación:** El Vendedor expone su catálogo de productos al Comprador con los datos relativos a cada producto: *nombre del producto, precio, fecha de expiración, etc.*
- **DescribirProducto:** Dado el nombre del producto buscado por el Comprador y si el Vendedor dispone de éste en su catálogo, le presenta los datos propios del producto, es decir: su *nombre, precio, fecha de expiración, cantidad en existencias, etc.* Los parámetros iniciales del agente Vendedor se corresponden con los datos del *producto*, de la siguiente manera: el *precio inicial* y el *precio deseado* son el precio de venta del producto; el *precio mínimo* de venta es el coste del producto, la *fecha final* de venta es la fecha de expiración del producto y la *fecha inicial* coincide con la fecha de elaboración del producto.
- **Regatear:** Efectúa el proceso de regateo mientras el precio del Comprador sea menor al *precio deseado* o al *precio mínimo* del agente Vendedor o mientras el *producto* no expire. El precio a ofertar se calcula usando la *función de decremento* explicada anteriormente.
- **Rechazar:** En cualquier momento, el Vendedor puede rechazar (no aceptar) el precio ofertado por el agente Comprador.
- **Vender:** Si el *precio* del Comprador es mayor o igual al *precio* ofertado por el Vendedor, entonces se realiza la venta.

- **Salir:** En cualquier momento, el agente Vendedor puede retirarse del Mercado Electrónico.

4.4.2 Lenguaje de Comunicación

La interacción entre el agente Comprador y el agente Vendedor se realiza siguiendo diferentes reglas de negociación que incluyen normas para admitir a los participantes en el Mercado Electrónico, para elegir el producto a negociar, para validar las propuestas, para efectuar propuestas, para rechazar propuestas, para hacer peticiones, para actualizar información de los participantes, para hacer conocer el estado de la negociación, para determinar si se ha llegado a un acuerdo final, para abandonar el mercado, etc. Dicha interacción se concreta usando un lenguaje simbólico de negociación a través de una pizarra electrónica. El mencionado lenguaje de negociación incluye un conjunto de primitivas que se describen a continuación:

- **ListoProducto(*t, prod*):** El Vendedor muestra los productos *prod* de la tienda *t*
- **DescripcionProducto(*prod*):** El Vendedor muestra las características del producto *prod*
- **Precio(*prod, pre*):** El Vendedor devuelve el precio *pre* del producto *prod*
- **Vendo(*prod, pre*):** El Vendedor oferta el producto *prod* al precio *pre*
- **TratoHecho(*prod, pre*):** Acepta el producto *prod* al precio *pre*
- **Rechazo(*prod, pre*):** No acepta el producto *prod* al precio *pre*
- **Salir(*prod*):** El agente se retira del mercado
- **BuscoProducto (*t, prod*):** El Comprador busca el producto *prod* en la tienda *t*
- **ElijoProducto (*prod, pre*):** El Comprador elige el producto *prod* al precio *pre*
- **Ofrezco(*prod, pre*):** El Comprador ofrece el precio *pre* por el producto *prod*

Un ejemplo de orden de ejecución de estas primitivas de negociación es el siguiente: el agente Comprador pregunta al agente Vendedor si tiene el producto que él desea (*BuscoProducto*). Posteriormente el Vendedor le responde diciendo si tiene o no el producto (*DescripcionProducto*). También, el Vendedor puede informar del precio del

producto (*Precio*). A continuación, el Comprador le puede ofertar una cantidad determinada (*Ofrezco*). El Vendedor puede contestar aceptando el precio ofertado (*TratoHecho*), no aceptando el precio ofertado (*Rechazo*), ofreciendo el producto a otro precio (*Vendo*) o retirándose del Mercado (*Salir*). El regateo puede continuar hasta que se acabe el tiempo límite del Comprador o del Vendedor o cuando uno de los dos agentes abandone el Mercado.

4.4.3 Pizarra de Comunicación

La implantación de comunicación entre el agente Comprador y el agente Vendedor se ha realizado mediante una pizarra electrónica, donde ambos pueden escribir y leer primitivas de negociación (ver Figura 12). Dicha pizarra incluye los siguientes datos:

- **PrimitivaVendedor:** Primitiva emitida por el Vendedor
- **PrecioVendedor:** Precio ofertado por el Vendedor
- **TiempoVendedor:** Fecha en la que se ha registrado la primitiva

- **PrimitivaComprador:** Primitiva emitida por el Comprador
- **PrecioComprador:** Precio ofertado por el Comprador
- **TiempoComprador:** Fecha en la que se ha registrado la primitiva

La pizarra funciona de la siguiente forma: el Vendedor escribe una primitiva de negociación en la pizarra. En las ranuras correspondientes al Vendedor se registran: la primitiva de negociación, el precio que se está ofertando y el instante de tiempo en que ocurre este evento. Similar situación ocurre cuando el Comprador quiere escribir una primitiva de negociación. Es de notar que sólo se registra la última primitiva emitida por el Vendedor (o por el Comprador). En el futuro se piensa ampliar la capacidad de esta pizarra de comunicación, de manera que se guarden en una cola todas las primitivas emitidas por los agentes. En el caso de lectura, ambos agentes pueden leer los datos de su adversario en cualquier momento.

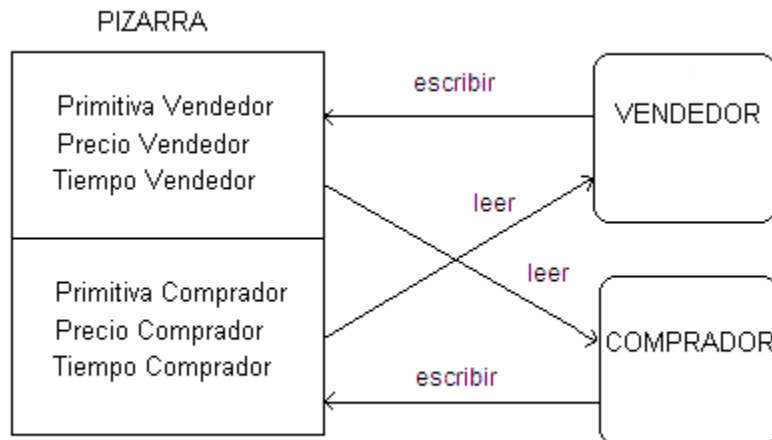


Figura 12. Pizarra electrónica de SNE

4.4.4 El Mercado Electrónico

El Mercado Electrónico es la plataforma donde pueden interactuar los agentes Comprador y Vendedor con el objetivo de comprar o vender productos al mejor precio posible. El Mercado funciona de la siguiente manera:

El agente Comprador, que representa al Cliente, ingresa al Mercado con los siguientes datos:

- *Nombre del producto*: El producto que le interesa comprar
- *Precio inicial*: El precio con el que comenzará el regateo del producto
- *Precio deseado*: El precio que le gustaría pagar por el producto
- *Precio máximo*: El precio máximo a pagar por el producto
- *Fecha inicial*: La fecha para empezar la búsqueda y compra del producto.
Se asume la fecha actual.
- *Fecha límite*: Fecha tope para comprar el producto

El agente Vendedor, que representa a la Tienda Virtual, ingresa al Mercado con los siguientes datos iniciales:

- *Nombre del producto*: Producto que quiere vender
- *Precio inicial*: El precio máximo con el que comienza la negociación. Es el coste del producto más un porcentaje de venta (por ejemplo: 25%).
- *Precio deseado*: El precio en el que le gustaría vender el producto. Coincide con el precio máximo de venta.
- *Precio mínimo*: El precio mínimo en el que se puede vender el producto. Es el coste del producto más un porcentaje de venta mínimo (por ejemplo: 10%).
- *Fecha inicial*: Se asume la fecha de creación o elaboración del producto
- *Fecha límite*: Coincide con la fecha de expiración del producto

Estos datos del Comprador y del Vendedor sólo sirven para inicializar los parámetros de compra y venta de ambos y no son conocidos por el adversario.

A continuación, se tiene dispuesto que una de las dos situaciones ocurra en el Mercado:

- a) El Comprador se aproxima a la Tienda Virtual y el Vendedor le presenta todos los productos disponibles de la Tienda (*ListoProductos*).
- b) Si existe el producto deseado, el Comprador señala su preferencia (*ElijoProducto*). Entonces, el Vendedor le da su precio (*Precio*). Si no existe el producto, el Comprador abandona el Mercado Electrónico (*Salir*).

O el siguiente escenario:

- a) El Comprador se aproxima a la Tienda Virtual buscando un producto determinado y pregunta al Vendedor si dispone de dicho producto (*BuscoProducto*).
- b) El Vendedor responde a la consulta de manera afirmativa o negativa, esto es, si existe el producto le da su precio (*Precio*). Si no existe el producto, se termina el proceso (*Salir*).

A continuación, el agente Comprador puede ejecutar una de las siguientes acciones, cada vez que le toque su turno. Cabe hacer notar que, las diferentes alternativas señaladas abarcan los distintos tipos de comportamientos mencionados previamente.

- Presenta su oferta al Vendedor, es decir, emite una propuesta de precio calculado según la función de incremento en vigencia (*Ofrezco*).
- Puede rechazar el precio ofertado por el Vendedor. Este mensaje de rechazo no va acompañado de ningún precio, más bien significa que prefiere una contraoferta del Vendedor menor que la actual (*Rechazo*).
- Acepta el precio ofertado por el Vendedor, por lo tanto se produce el “trato hecho” entre Comprador y Vendedor, terminando de manera exitosa el proceso de negociación del producto (*TratoHecho*).
- En cualquier momento, el Comprador puede abandonar el Mercado, con lo que se suspende el proceso de negociación del producto (*Salir*).

Asimismo, el agente Vendedor puede ejecutar una de las siguientes acciones cada vez que le corresponda negociar:

- Presenta su oferta al agente Comprador, es decir, emite una propuesta de precio que será calculado con la función de decremento vigente (*Vendo*).
- Puede rechazar el precio ofertado por el Comprador. Este mensaje de rechazo no va acompañado de ningún precio, más bien significa que quiere una contraoferta del Comprador más alta (*Rechazo*).
- Acepta el precio ofertado por el Comprador, por lo tanto se produce el “trato hecho” entre Comprador y Vendedor, terminando de manera exitosa el proceso de compra y venta del producto (*TratoHecho*).
- En cualquier momento, el Vendedor puede salirse del mercado, con lo que se termina el proceso de negociación del producto (*Salir*).

El conjunto de pasos anteriormente citados se repite mientras se cumplan las siguientes condiciones:

- Exista la “zona de acuerdo”.
- Los involucrados dispongan de tiempo para efectuar la compra o venta.
- Los agentes no abandonen el proceso de negociación.

4.5 Diseño del Sistema de Negociación Electrónica

El prototipo de SNE (Sistema de Negociación Electrónica) se implementó usando el lenguaje Java 1.6.0 y el editor BlueJ versión 2.1.3. Se usó programación orientada a objetos para esta construcción, como se puede apreciar en el diagrama de clases de la Figura 13.

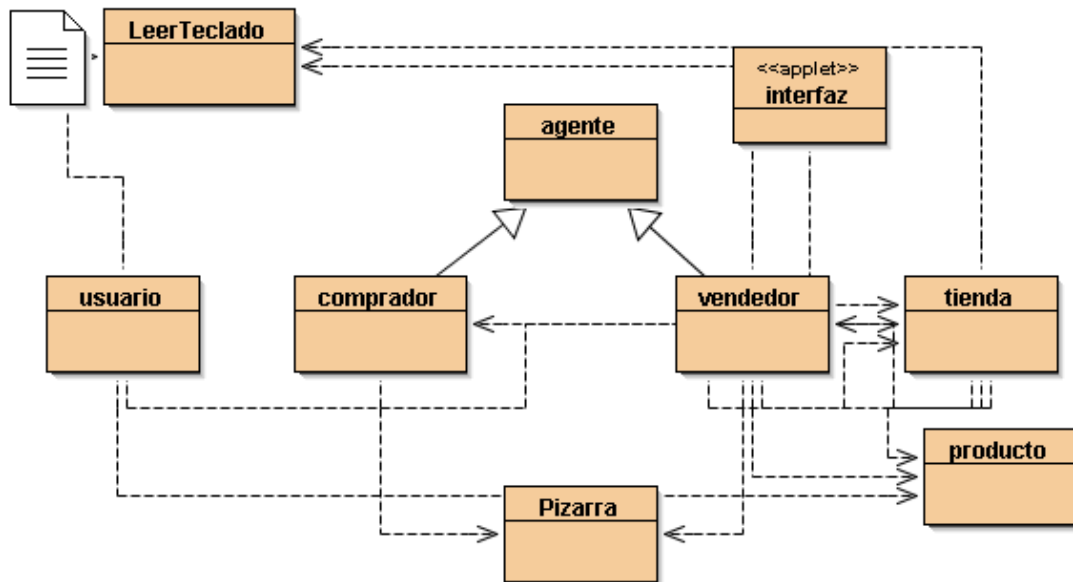


Figura 13. Diagrama de clases de SNE

Cada uno de los componentes de la negociación electrónica descritos en el apartado 4.4 se diseñó como un objeto: el agente Comprador y el Vendedor heredan de una plantilla general de Agente, con los atributos y comportamientos señalados en 4.4.1. El agente

Vendedor representa a la Tienda Virtual y a su vez, la Tienda está compuesta de Productos. El agente Comprador representa al Usuario. Las Primitivas de Comunicación se han implementado como métodos de los Agentes, que se llaman cuando ellos se quieren comunicar. La Pizarra se usa como medio de comunicación entre los Agentes, tal como se indicó arriba en 4.4.3. En esta implementación, dentro de la clase Interfaz, en el método *main()* están incluidos el algoritmo y las reglas de negociación del Mercado Electrónico.

4.6 Ejemplo de Funcionamiento del Sistema de Negociación Electrónica

Para ilustrar el funcionamiento del Sistema de Negociación Electrónica describiremos el funcionamiento de un Mercado Electrónico donde se pueden comprar y vender libros. En este ejemplo participarán dos agentes: un agente Comprador en representación del cliente y un agente Vendedor en representación de la Tienda Virtual de libros. La Tienda Virtual dispone de un conjunto de libros en su catálogo. Los libros tienen como atributos: nombre, precio, cantidad en almacenes, fecha de elaboración (en este caso, debería ser fecha de edición, pero la plantilla se ha hecho de manera general, para cualquier producto) y fecha de caducidad, principalmente (ver Figura 14). También se incluyen los atributos: coste, porcentaje de venta y porcentaje mínimo de venta, para ilustrar.

Producto: Libro
<i>Nombre:</i> "Maquiavelo"
<i>Precio:</i> 1000
<i>Cantidad:</i> 2
<i>Fecha Elaboración:</i> 01 / 04 / 2008
<i>Fecha Caducidad:</i> 04 / 10 / 2010
<i>Coste:</i> 800
<i>Porcentaje Venta:</i> 0.25
<i>Porcentaje Mínimo de Venta:</i> 0.1

Figura 14. Atributos del libro "Maquiavelo"

Los agentes Comprador y Vendedor ingresan al Mercado e informan acerca del producto que quieren comprar o vender. El agente Comprador quiere comprar el libro cuyo autor es "Maquiavelo" y el agente Vendedor, dispone de ese producto en la Tienda y puede

negociar por él. La *fecha inicial* del Comprador coincide con la fecha de inicio de la negociación. Entonces, cada uno empieza el proceso de negociación con los parámetros indicados en la Figura 15.

Agente Comprador
<i>Nombre Producto:</i> “Maquiavelo”
<i>Precio Inicial:</i> 850
<i>Precio Deseado:</i> 900
<i>Precio Máximo:</i> 1020
<i>Fecha Inicial:</i> 24 / 08 / 2008
<i>Fecha Límite:</i> 01 / 09 / 2008

Agente Vendedor
<i>Nombre Producto:</i> “Maquiavelo”
<i>Precio Inicial:</i> 1000
<i>Precio Deseado:</i> 1000
<i>Precio Mínimo:</i> 880
<i>Fecha Inicial:</i> 01 / 04 / 2008
<i>Fecha Límite:</i> 04 / 10 / 2010

Figura 15. Parámetros iniciales de los agentes Comprador y Vendedor

En la Figura 16 se puede apreciar el proceso de negociación seguido por los agentes que se muestra a los usuarios en las pantallas. Dicho reporte se expresa en lenguaje natural. La primera interacción ocurre en la fecha i ; la siguiente interacción ocurre en la fecha $i+1$ y así sucesivamente para que las funciones de incremento o decremento puedan calcular el nuevo precio a ofertar. En este ejemplo, se puede ver que en cuatro interacciones los agentes han llegado a un acuerdo en el precio del libro. También se puede acotar que, aunque no se muestra en el reporte, ambos agentes usaron la función de incremento (decremento) con el menor porcentaje de incremento (decremento) porque llegaron a un acuerdo en el primer periodo de tiempo de negociación (el primer periodo comprende del 24 al 28 de agosto, el segundo comprende del 29 al 30 de agosto y el tercer periodo del 31 de agosto al 1 de septiembre).

Fecha	Diálogo
24 / 08 / 2008	Comprador: Estoy buscando el producto “Maquiavelo”
24 / 08 / 2008	Vendedor: Tengo el producto “Maquiavelo” al precio de 1000. Indique su contra-oferta.
25 / 08 / 2008	Comprador: Oferto la cantidad de: 871.25
25 / 08 / 2008	Vendedor: Le vendo en el precio de 975
26 / 08 / 2008	Comprador: Oferto la cantidad de 914.81
26 / 08 / 2008	Vendedor: Le vendo en el precio de 926.25
27 / 08 / 2008	Comprador: Trato hecho. Acepto el precio de 926.25
27 / 08 / 2008	Vendedor: Entonces, trato hecho.

Figura 16. Diálogo de negociación del libro “Maquiavelo”

En la fecha 27/ 08/ 2008 el Comprador había calculado pagar 983.42, pero como el Vendedor le ofertó 926.25, el Comprador acepta este valor porque es más conveniente para él. Complementando, en la Figura 17, se muestra el conjunto de primitivas que los agentes usaron en el ejemplo para comunicarse dentro del Mercado Electrónico.

Fecha	Diálogo
24 / 08 / 2008	Comprador: BuscarProducto(Maquiavelo, 850)
24 / 08 / 2008	Vendedor: DescribirProducto(Maquiavelo, 1000)
25 / 08 / 2008	Comprador: Oferto(Maquiavelo, 871.25)
25 / 08 / 2008	Vendedor: Vendo(Maquiavelo, 975)
26 / 08 / 2008	Comprador: Oferto(Maquiavelo, 914.81)
26 / 08 / 2008	Vendedor: Vendo(Maquiavelo, 926.25)
27 / 08 / 2008	Comprador: TratoHecho(Maquiavelo, 926.25)
27 / 08 / 2008	Vendedor: TratoHecho(Maquiavelo, 926.25)

Figura 17. Primitivas de comunicación del libro “Maquiavelo”

También, puede darse el caso, que no haya acuerdo, ya sea porque no llegan a ningún arreglo económico o porque se acabe el plazo de negociación, como se puede ver en el siguiente ejemplo de la negociación de un libro de “Vargas Llosa” (ver Figura 18).

Producto: Libro
<i>Nombre:</i> “Vargas Llosa” <i>Precio:</i> 625 <i>Cantidad:</i> 5 <i>Fecha Elaboración:</i> 01 / 10 / 2007 <i>Fecha Caducidad:</i> 25 / 11 / 2008 <i>Coste:</i> 500 <i>Porcentaje Venta:</i> 0.25 <i>Porcentaje Mínimo de Venta:</i> 0.1

Figura 18. Atributos del libro “Vargas Llosa”

Los agentes Comprador y Vendedor ingresan al Mercado e informan acerca del producto que quieren comprar o vender. En esta ocasión el Vendedor despliega la lista de libros que dispone para que el agente Comprador pueda elegir el que desee. El agente Comprador elige el libro cuyo autor es “Vargas Llosa”, con lo que se inicia el proceso de negociación. En la Figura 19 se muestran los parámetros iniciales de negociación de ambos agentes.

Agente Comprador
<i>Nombre Producto:</i> “Vargas Llosa” <i>Precio Inicial:</i> 400 <i>Precio Deseado:</i> 460 <i>Precio Máximo:</i> 550 <i>Fecha Inicial:</i> 28 / 08 / 2008 <i>Fecha Límite:</i> 30 / 08 / 2008

Agente Vendedor
<i>Nombre Producto:</i> “Vargas Llosa” <i>Precio Inicial:</i> 625 <i>Precio Deseado:</i> 625 <i>Precio Mínimo:</i> 550 <i>Fecha Inicial:</i> 01 / 10 / 2007 <i>Fecha Límite:</i> 25 / 11 / 2008

Figura 19. Parámetros iniciales de los agentes Comprador y Vendedor

En la Figura 20 se puede apreciar el proceso de negociación seguido por los agentes en este segundo ejemplo. Se puede ver que aunque ocurren cuatro interacciones, no se llega a ningún acuerdo porque se le acaba el plazo al Comprador.

Fecha	Diálogo
28 / 08 / 2008	Vendedor: Tengo estos productos. ¿Cuál te interesa? “Vargas Llosa”, precio: 625 “Mesa Gisbert”, precio: 1562.5 “Maquiavelo”, precio: 1000 “García Márquez”, precio: 125 “Juan de la Rosa”, precio: 986.25
28 / 08 / 2008	Comprador: Me interesa el producto “Vargas Llosa” y oferto 400
28 / 08 / 2008	Vendedor: Precio 625. Indique su contra-oferta
29 / 08 / 2008	Comprador: Oferto la cantidad de 410
29 / 08 / 2008	Vendedor: Le vendo en el precio de 593.75
30 / 08 / 2008	Comprador: Oferto la cantidad de 430.5
30 / 08 / 2008	Vendedor: Le vendo en el precio de 534.375
31 / 08 / 2008	Comprador: Se me acaba el tiempo. Adiós.

Figura 20. Diálogo de negociación del libro “Vargas Llosa”

De igual forma, en la Figura 21, se muestra el conjunto de primitivas que los agentes usaron en el ejemplo de negociación de un libro de “Vargas Llosa”.

4.7 Análisis de Resultados

El objetivo de esta primera implementación es conseguir la estructura básica de Mercado Electrónico donde participen agentes software con capacidad de negociar el precio de un producto. En principio se verificó y validó cada uno de los componentes de este primer

prototipo desarrollado: agentes, lenguaje de comunicación, pizarra, mercado electrónico y el algoritmo de negociación, para comprobar su correcto funcionamiento.

Fecha	Diálogo
28 / 08 / 2008	Vendedor: ListoProductos()
28 / 08 / 2008	Comprador: ElijoProducto("Vargas Llosa", 400)
28 / 08 / 2008	Vendedor: Precio("Vargas Llosa", 625)
29 / 08 / 2008	Comprador: Oferto("Vargas Llosa", 410)
29 / 08 / 2008	Vendedor: Vendo("Vargas Llosa", 593.75)
30 / 08 / 2008	Comprador: Oferto("Vargas Llosa", 430.5)
30 / 08 / 2008	Vendedor: Vendo("Vargas Llosa", 534.375)
31 / 08 / 2008	Comprador: Salir()

Figura 21. Primitivas de comunicación del libro "Vargas Llosa"

Posteriormente, se condujeron cinco experimentos con cada usuario (en total 10 usuarios) para evaluar el funcionamiento del Mercado Electrónico, y en particular, el algoritmo de negociación. Se deseaba obtener la opinión de los usuarios del proceso de negociación propuesto y también evaluar qué tan buenos son los precios que puede obtener el agente Comprador. Si ese fuera el caso, también se indagaría si el usuario confiaría en el agente Comprador en procesos de negociación reales.

Los clientes ingresaron los parámetros iniciales con los que querían que negociara el agente Comprador, esto es: *nombre del producto*, *precio inicial*, *precio deseado*, *precio máximo* y *fecha límite*. El sistema asume que la *fecha inicial* es la fecha en la que empieza el proceso de negociación. Los valores de los parámetros iniciales del agente Vendedor, esto es: *nombre del producto*, *precio inicial*, *precio deseado*, *precio mínimo*, *fecha inicial* y *fecha límite* son dados por los valores del producto a vender, como se explicó en 4.4.4. A continuación, y por pantalla, los clientes humanos pudieron observar como se iba dando la negociación, como se mostró en los ejemplos del apartado 4.6.

De acuerdo a los 50 ejemplos ingresados, en promedio, el 50% de los ejemplos llegaron a buen término y el otro 50% no. Por lo que se comprueba que estadísticamente, el algoritmo de negociación funciona correctamente. En cuanto a la opinión de los usuarios acerca del sistema, ellos dicen que aunque la interfaz del sistema es muy rústica, el comportamiento de los agentes es satisfactorio. Ellos comentaron que el algoritmo de negociación refleja de buena manera las negociaciones humanas, porque al principio el agente Comprador se muestra prudente a tiempo de presentar sus ofertas, pudiendo obtener el producto a un precio cercano a su precio mínimo y cerca al final de la negociación, presenta un comportamiento más audaz, pudiendo ofertar lo máximo posible.

Sin embargo, otros comportamientos no les convencieron mucho. A veces, el agente Comprador actúa sin tomar en cuenta el comportamiento del Vendedor, como cuando la diferencia de precios entre el Comprador y Vendedor es muy grande y el Comprador incrementa su oferta de manera tan escasa, que su comportamiento se advierte descontextualizado. En otro punto, los usuarios observaron que el adversario podría extrapolar el comportamiento del Comprador dadas las funciones matemáticas que se usan. Por eso se incluyó el valor aleatorio en las funciones (ver 4.4.1), para desorientar al opositor en la medida de lo posible.

Acerca de los precios que consigue el agente Comprador, los usuarios se mostraron satisfechos y asombrados por los resultados, sobre todo porque a las personas, en general, les desagrada regatear y cuando lo hacen, se conforman con el primer precio razonable que consiguen [15]. En cambio, el agente, puede agotar todas las posibilidades. Los usuarios se mostraron felices con la idea de que existieran herramientas de este tipo, pero en cuanto se les planteó la posibilidad de delegar esta tarea en aplicaciones reales, ellos dudaron bastante. Es natural, porque nadie quiere arriesgar su dinero.

5. Negociación Automática con Aprendizaje

5.1 Introducción

El propósito de este trabajo es desarrollar agentes software que se desenvuelvan en un mercado virtual y que tengan capacidad de negociar el precio de un producto en representación de sus clientes, de manera que actúen como sus asistentes personales. Para lograr esto, los agentes software deberán contar con capacidad de aprendizaje para adquirir paulatinamente las destrezas de negociación de sus clientes.

5.2 Análisis de los Algoritmos de Aprendizaje

En general, en un proceso de negociación, tanto el comprador como el vendedor, se encuentran en una contienda donde tiene que haber un ganador. Dentro de la contienda, cada uno decide cuándo ingresar al mercado, cuándo y cuánto ofertar, cuándo rechazar la contraoferta, cuándo aceptarla, cuándo salir del mercado, siempre tomando en cuenta el contexto actual, y si es posible, las situaciones pasadas. Normalmente, los adversarios tienen poca información de sus oponentes y la van conociendo a medida que se desarrolla la negociación. Además, un buen vendedor, o comprador, se hacen con el tiempo. Ellos van acumulando en su haber una serie de negociaciones pasadas, tanto exitosas como fallidas, que les servirán como referente para las futuras negociaciones.

Los agentes del Sistema de Negociación Electrónica (SNE) desarrollado ya tienen características que les permiten negociar dentro de un Mercado Electrónico, pero carecen de capacidad de aprendizaje que les facilite asimilar la manera de negociar de sus clientes y en futuras ocasiones, aprovechar dicho conocimiento para obtener mejores resultados.

En el capítulo 2 se hizo un estudio exhaustivo de diferentes algoritmos de aprendizaje. Algunos de ellos sólo representan situaciones puntuales y no un contexto completo al tiempo de tomar decisiones, por ejemplo: la reglas de decisión, en sus diferentes

versiones de lógica proposicional o lógica de primer orden. Es verdad que podrían hacerlo, pero habría que encadenar las reglas al estilo de un sistema experto tradicional. Otros algoritmos sirven para clasificar objetos, entre ellos los árboles de decisión y las redes neuronales, pero lo que se quiere es representar el proceso de negociación y utilizarlo en posteriores negociaciones. Los algoritmos de agrupamiento permiten reunir objetos por su parecido, pero este no es el problema que se desea resolver. Los algoritmos de refuerzo permiten saber si el conjunto de pasos que ha seguido el agente están bien o no, pero no registran todo el conjunto de acciones realizadas por el agente. Los algoritmos evolutivos consiguen obtener resultados óptimos cuando disponen de los datos completos del problema en cuestión, pero en el presente trabajo no se cuenta con los datos del adversario.

De esta forma, se seleccionó la técnica de Razonamiento Basado en Casos (RBC) para representar cada proceso de negociación como un caso o situación. El agente software progresivamente almacenará en su memoria los procesos de negociación realizados, las adaptará a las nuevas negociaciones y poco a poco se debería volver diestro en conseguir buenos acuerdos. Consiguientemente, se obtendría un algoritmo de aprendizaje incremental e interactivo en procesos de negociación electrónica.

En este trabajo se supone que los agentes software no conocen los valores de reserva de su adversario (precios y fechas tope), tienen un comportamiento confiable, siempre dicen la verdad y no son mal intencionados.

5.3 Arquitectura del Sistema de Aprendizaje de Negociación Electrónica

En la Figura 22 se pueden apreciar los cambios que se han hecho al primer prototipo de Negociación Electrónica. En esta oportunidad se puede ver que se ha añadido inteligencia sólo al agente Comprador, con el propósito de evaluar los beneficios que le aporta la capacidad de aprendizaje en el proceso de negociación versus el Vendedor sin inteligencia.

El agente Comprador usará Razonamiento Basado en Casos para efectuar el proceso de negociación, por lo que su comportamiento actual dependerá de casos anteriores exitosos, aplicando el principio de: “las situaciones similares son situaciones cuyos resultados también lo serán” [1]. El Comprador dispone de una Memoria de Casos que se irá alimentando a medida que pase el tiempo y de un módulo de Gestión de Casos que le permitirá la recuperación, re-uso y almacenamiento de los casos.

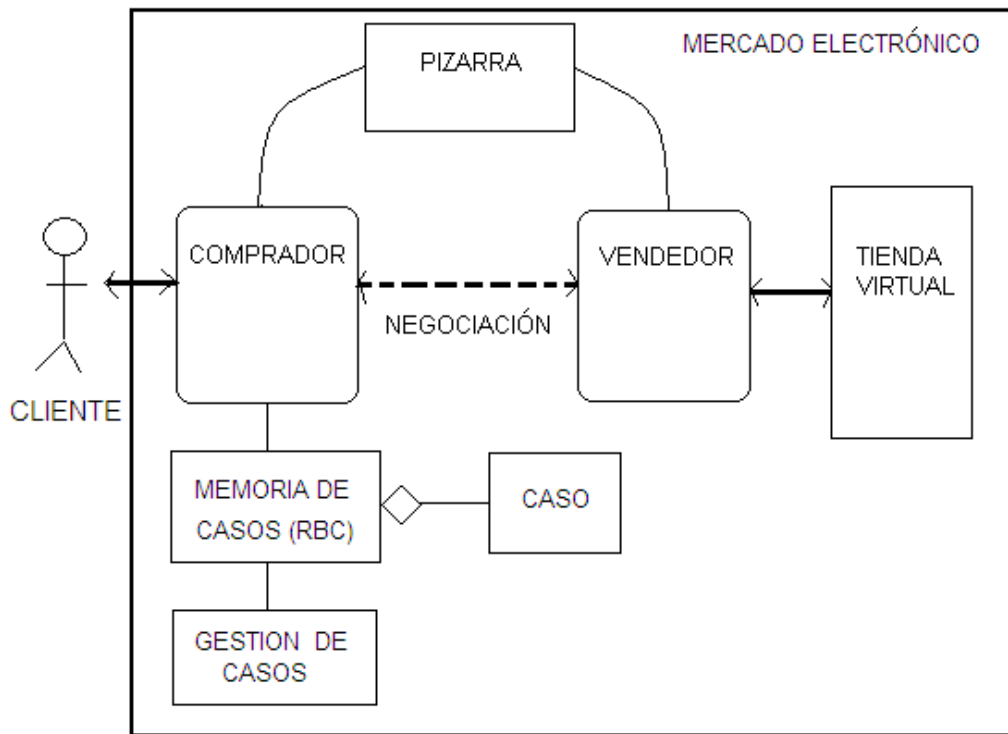


Figura 22. Arquitectura del Sistema de Aprendizaje de Negociación Electrónica (SANE)

5.3.1 Agente Comprador Inteligente

Cuando se resuelve un problema usando Razonamiento Basado en Casos (RBC) se debe emparejar el caso actual con alguno de la base de casos, debiendo, por lo tanto, recuperarse casos similares. Para permitir la recuperación de casos cuyos compradores se parezcan al agente Comprador se han añadido los siguientes atributos:

- *interesProductoC*: Grado de interés por el producto (por ejemplo, 1: poco, 2: regular, 3: bastante, 4: mucho). Se supone que si el Comprador tiene más interés por el producto, entonces invertirá más dinero para obtenerlo.
- *nivelAdquisitivoC*: Nivel adquisitivo medido en unidades monetarias (por ejemplo, 1: 0-500, 2: 501-1500, 3:1501-10000, 4: 10000-∞). Si el Comprador tiene mayor poder adquisitivo, entonces existirán más posibilidades de que compre del producto.

Los atributos como *precio inicial* y *precio máximo* a pagar por el producto y *fecha inicial* y *fecha límite* del proceso de negociación del Comprador, también se usarán para permitir la recuperación de sujetos parecidos.

- *precioInicialC* y *precioMaximoC*: Rango de precios que desea pagar el Comprador.
- *fechaInicialC* y *fechaLimiteC*: Fecha de inicio y fin del proceso de negociación del Comprador.

Con estos atributos se pueden recuperar casos similares por diversas condiciones, por ejemplo: se pueden recuperar compradores parecidos en nivel adquisitivo, en interés por el producto, en periodos de tiempo en los que quisiera adquirir el producto o en intervalos de precio deseados, tomando en cuenta una o todas las condiciones.

5.3.2 Casos de SANE

Los casos implantados en SANE deberían incluir los datos más relevantes del proceso de negociación. Se vio por conveniente incluir tres tipos de conjunto de datos: los datos generales de la negociación y los datos particulares de cada agente.

La estructura de un caso de SANE tiene la siguiente forma:

- Datos generales:** Los datos generales de un caso servirán para disponer de los datos globales de la negociación: el producto que se negoció, las fechas en las que

ocurrió la negociación, el periodo de tiempo en que ocurrió la negociación, si la negociación tuvo éxito o no, si el caso se reutilizó o no y cuántas veces.

- *idCaso*: Identificador del caso
- *Producto*: Nombre del producto a negociar
- *fechaInicial* y *fechaFinal*: Fechas de inicio y fin de la negociación
- *precioInicial* y *precioFinal*: Precios inicial y final de la negociación
- *tratoHecho*: Si se ha efectuado la compra y venta del producto
- *seSalio*: Indica qué agente abandonó el proceso de negociación
- *nroVecesUsado*: Número de veces que se ha usado el caso
- *nroVecesUsadoExito*: Número de veces que se ha usado con éxito

b) Datos del agente Vendedor: Los datos principales del agente Vendedor son el conjunto de pares: precio – fecha. Las fechas son aquellas en las que ocurren las interacciones con el Comprador y los precios son los que se ofertan en dichas fechas.

- *nombreVendedor*: Nombre del agente Vendedor
- *listaPreciosV*: Conjunto de precios que ha ofertado durante la negociación
- *listaFechasV*: Conjunto de fechas en las que ha ofertado
- *listaPrimitivasV*: Conjunto de primitivas de comunicación que ha enviado

c) Datos del agente Comprador: El agente Comprador incluye los atributos anteriormente mencionados en el punto 5.3.1 para permitir su recuperación cuando se aplique el Razonamiento Basado en Casos. También incluye el conjunto de pares: precio – fecha. Las fechas son aquellas en las que ocurren las interacciones con el Vendedor y los precios son los que se ofertan en dichas fechas. Este conjunto de pares precio – fecha sirven para construir la función matemática *funcionC* que les representará y que tiene como fin pronosticar los precios que podrá ofertar en las nuevas negociaciones.

- *nombreComprador*: Nombre del agente Comprador.
- *interesProductoC*: Grado de interés por el producto (por ejemplo, 1: poco, 2: regular, 3: bastante, 4: mucho).
- *nivelAdquisitivoC*: Nivel adquisitivo medido en unidades monetarias (por ejemplo, 1: 0-500, 2: 501-1500, 3:1501-10000, 4: 10000-∞).
- *precioInicialC* y *precioMaximoC*: Rango de precios que desea pagar por el producto.
- *fechaInicialC* y *fechaLimiteC*: Periodo de tiempo en el que desea efectuar la negociación. Se mide en número de días.
- *listaPreciosC*: Conjunto de precios que ha ofertado durante la negociación.
- *listaFechasC*: Conjunto de fechas en las que ha ofertado
- *listaPrimitivasC*: Conjunto de primitivas de comunicación que ha enviado
- *funcionC*: Función matemática obtenida por un proceso de regresión estadística a partir de *listaPreciosC* y *listaFechasC*. Puede ser: lineal, logarítmica, parabólica, potencial, exponencial o hiperbólica.
- *errorC*: Porcentaje de error de la *funciónC* en relación a los datos usados.

5.3.3 Memoria de Casos

Una de las características del Razonamiento Basado en Casos es asumir que los problemas nuevos son similares a los previos, y que por lo tanto, las soluciones anteriores se pueden reutilizar, ya sea de forma directa o por algún proceso de adaptación. Por eso, es importante disponer de una base o memoria de casos. En SANE se determinaron dos tipos de casos, por lo tanto se debe disponer de dos *memorias de casos*:

- a) *Memoria de casos positivos*: Casos en los que se ha llegado a feliz término en el proceso de compra y venta de productos.
- b) *Memoria de casos negativos*: conjunto de casos en los que el proceso de compra y venta de productos no ha llegado a su finalización exitosa (por ejemplo: debido al retiro de los agentes del mercado electrónico, al concluirse el tiempo permitido, etc.)

Los *casos* en SANE se implementaron como objetos del lenguaje Java, cuyos atributos fueron descritos en el punto 5.3.2. Aprovechando las estructuras de datos pre-construidas y los métodos que incluye el lenguaje Java, las *memorias de casos* de SANE, positiva y negativa, se implementaron de dos formas:

- De forma permanente los casos se almacenan con su estructura original, esto es, como objetos. Se usan los métodos de serialización y deserialización del lenguaje Java para escribir y leer objetos en archivos binarios. El número de casos que se han manejado en este segundo prototipo no ha superado los 500, por lo que no fue necesario usar gestores de bases de datos (por ejemplo: MySQL) para optimizar este proceso.
- En tiempo de ejecución, los casos se almacenan en tablas Hash (Hashtable). Se deserializan los casos almacenados en los ficheros binarios, se los inserta en las tablas Hash y posteriormente se realizan las tareas de gestión pertinentes de SANE de forma muy eficiente. Cuando se termina de usar SANE, los casos de las tablas Hash se serializan y se guardan en los ficheros binarios de Java.

5.3.4 Gestión de Casos

En SANE la gestión de casos incluye las siguientes tareas:

5.3.4.1 Creación

Cada *caso*, con la estructura señalada en el apartado 5.3.2 se crea como un objeto de Java y sus atributos toman valores a medida que ocurre el proceso de negociación de los agentes en el mercado electrónico (por ejemplo: precios, primitivas de comunicación, fechas, etc.). Los *casos* tienen como identificador e índice a un número secuencial que se va incrementando a medida que se crean más casos. Después de la creación y llenado de valores, el *caso* se inserta en la correspondiente tabla Hash del Sistema, para posteriormente serializarse en los archivos binarios o *memoria de casos*.

5.3.4.2 Recuperación

Cuando un agente Comprador usa Razonamiento Basado en Casos, debe recuperar de su *memoria de casos*, el caso más parecido al *caso actual* para aprovechar el conocimiento previo. En SANE, en esta primera etapa, sólo se recuperan casos de la *memoria de casos positivos* para obtener conocimiento de negociaciones pasadas exitosas. Se implantaron dos tipos de recuperación de casos:

- **Recuperación del caso en que se haya comprado el producto p al menor precio:** Se recupera de la *memoria de casos positivos* el *caso* en el que el comprador haya regateado de mejor manera por el producto p , y por lo tanto, tenga el precio más bajo de todos los casos positivos.
- **Recuperación del caso más parecido:** Se recupera el *caso* cuyo *interés por el producto, nivel adquisitivo, precio inicial de compra, precio máximo de compra, fecha inicial de compra o fecha límite* para efectuar la compra sean más parecidos al *caso actual*. Esta recuperación se realiza usando el concepto de *vecino más próximo*, descrito en el apartado 2.2.5. Se aplica la fórmula de la distancia Euclídea entre el *caso actual* y los casos de la *memoria de casos positivos* y se recupera el *caso* cuya distancia sea menor con el *caso actual*.

El caso recuperado se denomina *caso similar*.

5.3.4.3 Adaptación

Después de recuperar el *caso similar*, por cualquiera de los dos métodos explicados anteriormente, el agente Comprador inteligente tiene los insumos suficientes para calcular el nuevo precio a ofertar. Tanto el *caso actual* como el *caso similar* incluyen una lista de precios y una lista de fechas del Comprador que representan el precio que ha ofertado el Comprador al Vendedor en determinada fecha. En general, las fechas del *caso*

similar y del *caso actual* no coincidirán, por lo que es necesario normalizarlas. Por este motivo, a ambas se las transforma de fechas a números que comienzan en el valor numérico: 1, 2, 3 y así sucesivamente.

A continuación, a partir del conjunto de pares precio – fecha normalizado del *caso similar* y aplicando técnicas de regresión estadística se obtiene la función matemática que modela el comportamiento de los precios en función de las fechas. En SANE se han implementado las siguientes funciones de regresión: lineal, logarítmica, parabólica, potencial, exponencial e hiperbólica; de manera que se obtiene la función que mejor se ajusta a los datos del *caso similar*.

Por ejemplo, si el conjunto de pares precio – fecha del *caso similar* se representa mejor con una función lineal, se consigue la siguiente ecuación:

$$\text{precio}_i^{\wedge} = \beta_0 + \beta_1 \text{fecha}_i + \varepsilon \quad (\text{Ecuación 4})$$

Donde precio_i^{\wedge} es el nuevo precio a calcular, β_0 y β_1 son los coeficientes obtenidos en la regresión lineal y ε es el error de la función calculada. Por lo tanto, el nuevo precio_i^{\wedge} a ofertar en la fecha_i se consigue aplicando la Ecuación 4. Así se consigue adaptar el comportamiento del *caso similar* en el *caso actual*.

5.3.4.4 Revisión y Almacenamiento

En SANE, cuando se termina el proceso de negociación, el *caso actual* se revisa y posteriormente se almacena en la *memoria de casos*. La revisión de un caso, en este Sistema, consiste en determinar si el caso es positivo o negativo para almacenarlo en la *memoria de casos* correspondiente. El proceso de almacenamiento de casos en SANE se explicó detalladamente en 5.3.3.

5.4 Diseño del Sistema de Aprendizaje de Negociación Electrónica

En la Figura 23 se puede apreciar el diagrama de clases de SANE, creado en el editor BlueJ para programación en Java. En este prototipo, respecto al presentado en 4.5, se puede observar que se han añadido clases relacionadas con el Razonamiento Basado en Casos tales como: Caso, NoCasos, ListaCasos, Aprendizaje y Regresión. La clase Caso implementa todos los atributos descritos en 5.3.2. En ListaCasos se tiene la tabla Hash que almacena los Casos en tiempo de ejecución. Lista Casos permite insertar, recuperar, eliminar y buscar casos en la tabla Hash. Los dos tipos de recuperación de casos (por precio y por parecido) se implementan en ListaCasos. Cuando se termina la ejecución de SANE, los casos se guardan en archivos binarios en el disco duro del ordenador y cuando empieza el funcionamiento de SANE, los casos se recuperan de los archivos binarios y se cargan a las tablas Hash respectivas.

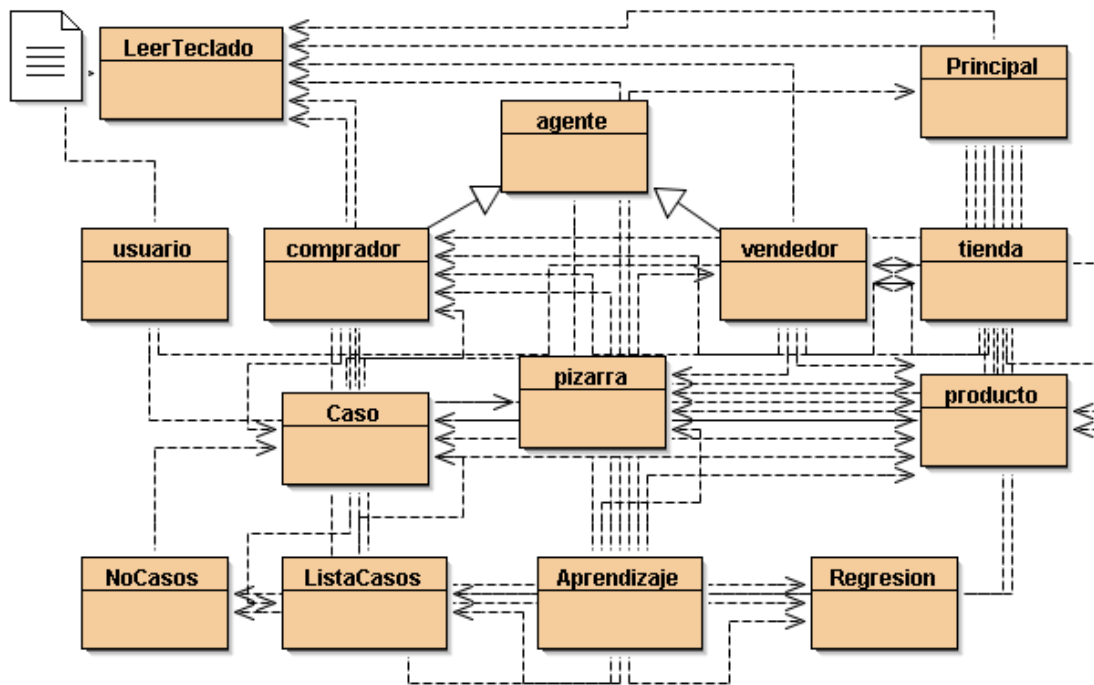


Figura 23. Diagrama de clases de SANE

La clase NoCasos almacena el número secuencial que identifica unívocamente a cada caso de SANE. Para mantener la secuencia, este número se guarda en un archivo binario en disco duro. La clase Regresión incluye todos los métodos necesarios para calcular diversas funciones de regresión estadística (por ejemplo: lineal, logarítmica, potencial, etc.). Finalmente, en la clase Aprendizaje está incluido el algoritmo de negociación entre el agente Comprador y el Vendedor, pero esta vez, el precio que oferta el agente Comprador se recupera y adapta de un Caso de ListaCasos.

5.5 Funcionamiento del Sistema de Aprendizaje de Negociación Electrónica

Para mostrar el funcionamiento del Sistema de Aprendizaje de Negociación Electrónico se usará el primer ejemplo descrito en 4.6, en el cual el agente Comprador desea adquirir el libro “Maquiavelo” y el agente Vendedor dispone de este libro. Pero en esta nueva situación, el agente Comprador incorpora Razonamiento Basado en Casos. En la Figura 24 se repiten los datos del libro que se va a negociar.

Producto: Libro
<i>Nombre:</i> “Maquiavelo”
<i>Precio:</i> 1000
<i>Cantidad:</i> 2
<i>Fecha Elaboración:</i> 01 / 04 / 2008
<i>Fecha Caducidad:</i> 04 / 10 / 2010
<i>Coste:</i> 800
<i>Porcentaje Venta:</i> 0.25
<i>Porcentaje Mínimo de Venta:</i> 0.1

Figura 24. Atributos del libro “Maquiavelo”

Los agentes Comprador y Vendedor empiezan el proceso de negociación con los parámetros indicados en la Figura 25.

Agente Comprador
<i>Nombre Producto:</i> “Maquiavelo”
<i>Precio Inicial:</i> 850
<i>Precio Deseado:</i> 900
<i>Precio Máximo:</i> 1020
<i>Fecha Inicial:</i> 24 / 08 / 2008
<i>Fecha Límite:</i> 01 / 09 / 2008

Agente Vendedor
<i>Nombre Producto:</i> “Maquiavelo”
<i>Precio Inicial:</i> 1000
<i>Precio Deseado:</i> 1000
<i>Precio Mínimo:</i> 880
<i>Fecha Inicial:</i> 01 / 04 / 2008
<i>Fecha Límite:</i> 04 / 10 / 2010

Figura 25. Parámetros iniciales de los Agentes Comprador y Vendedor

El Cliente puede elegir recuperar el *caso* en que el comprador se parezca más a él o recuperar el *caso* en que se haya comprado más barato. En la Figura 26 se muestra un ejemplo en el que se recupera el *caso* en el que se ha comprado el libro “Maquiavelo” al precio más bajo de toda la *memoria de casos positivos*.

Se puede observar que, en el caso recuperado, el Comprador obtuvo el libro “Maquiavelo” en el precio de 900, valor menor al conseguido por el Comprador sin aprendizaje del capítulo 4, que fue de 926.25. También se puede apreciar que el atributo *funcionC* tiene como valor una función parabólica que se ha obtenido a través de un proceso de regresión estadística. Esta función es la que mejor modela la relación entre los pares *listaPreciosC* y *listaFechasC*, puesto que su error estimado es de 1.9. Las otras funciones matemáticas que implementa SANE, tales como: lineal, logarítmica, potencia, exponencial e hiperbólica obtienen errores mayores a 1.9.

Caso
<i>idCaso</i> : 90 <i>Producto</i> : “Maquiavelo” <i>fechaInicial</i> : 10/ 07 / 2008 <i>fechaFinal</i> : 14/ 07 / 2008 <i>precioInicial</i> : 800 <i>precioFinal</i> : 900 <i>tratoHecho</i> : true <i>seSalio</i> : false
<i>listaPreciosV</i> : {1000, 975, 975, 926.25, 900} <i>listaFechasV</i> : {10/07/2008, 11/07/2008, 12/07/2008, 13/07/2008, 14/07/2008}
<i>interesProductoC</i> : 3 <i>nivelAdquisitivoC</i> : 4 <i>precioInicialC</i> : 800 <i>precioMaximoC</i> : 980 <i>fechaInicialC</i> : 10/ 07 /2008 <i>fechaLimiteC</i> : 20/ 07 /2008 <i>listaPreciosC</i> : {800, 820, 855, 900, 900} <i>listaFechasC</i> : {10/07/2008, 11/07/2008, 12/07/2008, 13/07/2008, 14/07/2008} <i>funcionC</i> : $\text{precio}^{\wedge} = 790.7 + 6.9 \text{ fecha}^2$ <i>errorC</i> : 1.9

Figura 26. Caso recuperado con el menor precio para el libro “Maquiavelo”.
Sólo se muestran los atributos indispensables.

Los siguientes precios que ofertará el agente Comprador se van a calcular usando dicha función parabólica. Por ejemplo, para la primera fecha, cuyo valor normalizado es 1, se calcula así:

$$\text{precio}^{\wedge} = 790.7 + 6.9 (1)^2$$

$$\text{precio}^{\wedge} = 799$$

Y así sucesivamente para las siguientes fechas.

En la Figura 27 se muestra el diálogo establecido entre ambas partes. También se observa que el Comprador, al adoptar el comportamiento del *caso recuperado* consigue un precio mucho mejor que el registrado en dicho *caso*. Como el Comprador empezó con un precio muy bajo, el Vendedor se vio obligado a rebajar su precio hasta llegar a ofertar el *precio mínimo* por el producto. El Comprador aprovechó esta situación y convino en el acuerdo. Según la función parabólica, en la fecha 27/ 08/2008, el próximo precio que hubiera

ofertado el Comprador sería de 901, pero al ver que el Vendedor le hace una oferta menor, acepta el precio de 880.

Fecha	Diálogo
24 / 08 / 2008	Vendedor: Tengo estos productos. ¿Cuál te interesa? “Vargas Llosa”, precio: 625 “Mesa Gisbert”, precio: 1562.5 “Maquiavelo”, precio: 1000 “García Márquez”, precio: 125 “Juan de la Rosa”, precio: 986.25
24 / 08 / 2008	Comprador: Me interesa el producto “Maquiavelo” y oferto 799
25 / 08 / 2008	Vendedor: Le vendo en el precio de 975
25 / 08 / 2008	Comprador: Le oferto la cantidad de 818.36
26 / 08 / 2008	Vendedor: Le vendo en el precio de 926.25
26 / 08 / 2008	Comprador: Le oferto la cantidad de 853
27 / 08 / 2008	Vendedor: Le vendo en el precio de 880
27 / 08 / 2008	Comprador: Trato hecho. Acepto el precio de 880
27 / 08 / 2008	Vendedor: Entonces, trato hecho.

Figura 27. Diálogo de negociación del libro “Maquiavelo”

En la Figura 28 se muestra el conjunto de primitivas que los agentes usaron en el ejemplo anterior.

5.6 Análisis de Resultados

El objetivo de esta segunda implementación es evaluar el impacto de incluir una técnica de Aprendizaje Automático en el Sistema de Negociación Electrónico. En esta oportunidad, se seleccionó el Razonamiento Basado en Casos por las razones explicadas anteriormente.

En principio, se evaluaron los nuevos módulos añadidos al Sistema (Regresión, Caso, Memoria de Casos y Aprendizaje) y se comprobó su correcto funcionamiento. El

Mercado Electrónico funciona de la misma manera, pero ahora, los precios que oferta el Comprador se calculan con base al *caso similar* recuperado de la *memoria de casos* de SANE y usando el proceso de adaptación explicado en 5.3.4.3.

Fecha	Diálogo
24 / 08 / 2008	Vendedor: ListoProductos()
24 / 08 / 2008	Comprador: ElijoProducto("Maquiavelo", 799)
25 / 08 / 2008	Vendedor: Precio("Maquiavelo", 975)
25 / 08 / 2008	Comprador: Oferto("Maquiavelo", 818.36)
26 / 08 / 2008	Vendedor: Vendo("Maquiavelo", 926.25)
26 / 08 / 2008	Comprador: Oferto("Maquiavelo", 853)
27 / 08 / 2008	Vendedor: Vendo("Maquiavelo", 880)
27 / 08 / 2008	Comprador: TratoHecho("Maquiavelo", 880)
27 / 08 / 2008	Vendedor: TratoHecho("Maquiavelo", 880)

Figura 28. Primitivas de comunicación del libro "Maquiavelo"

Se realizaron cinco experimentos con 10 usuarios para evaluar el funcionamiento del Mercado Electrónico con Aprendizaje. Los usuarios ingresaron los datos iniciales pertinentes y en especial: el *grado de interés por el producto*, el *nivel adquisitivo* y qué tipo de recuperación de casos preferían. Usando los mismos ejemplos que en el Mercado de Negociación sin Aprendizaje, los resultados obtenidos muestran que cuando el agente Comprador adopta el comportamiento de un *caso con el menor precio posible*, el 76% de las veces consigue mejores precios que cuando no usa el Razonamiento Basado en Casos. Cuando el agente Comprador adopta el comportamiento del *caso más parecido* los resultados no son tan alentadores, puesto que su eficacia es sólo del 57%, aunque mayor que en el Sistema sin aprendizaje. Esto es claro, porque un agente con más conocimiento tiene más pautas de cómo y cuánto ofertar para obtener el producto a un buen precio, aunque no siempre lo consiga.

Los usuarios se mostraron muy satisfechos con el desempeño de este segundo prototipo, porque consigue mejores resultados que el anterior prototipo y también porque consigue mejores resultados que ellos mismos. Este hecho nos demostró que se puede llegar a

conseguir a implantar un sistema de negociación automática para el comercio electrónico B2C que satisfaga la expectativa de las personas.

La implantación del Razonamiento Basado en Casos en SANE permite contar con un algoritmo de aprendizaje interactivo e incremental, que con el tiempo va mejorando su desempeño. Como la *memoria de casos* se incrementará, próximamente se evaluará cómo optimizar la representación y el almacenamiento de los casos, pudiendo usar un gestor de base de datos.

También, las siguientes evaluaciones de SANE se realizarán usando las medidas de desempeño de negociación mencionadas en 3.4.2, tales como: Conjunto de beneficios/utilidad, balance de contrato, distancia a la frontera eficiente y distancia a la solución de regateo Nash, para tener una valoración objetiva del Sistema

Finalmente, al realizar las pruebas se consiguieron algunas heurísticas para mejorar el desempeño del agente Comprador, como por ejemplo:

- Si el comprador empieza su cotización lo más baja posible, entonces puede conseguir un mejor trato (obliga al vendedor a ceder en su precio).
- Si los incrementos del comprador son menores, entonces puede conseguir un mejor precio.

Dichas heurísticas también podrían implantarse en SANE como reglas de decisión. También se ha pensado en incluir otras técnicas inteligentes para conseguir que el comportamiento de los agentes sea más cercano al humano.

6. Conclusiones y Trabajo Futuro

6.1 Conclusiones

En este proyecto se ha desarrollado un Sistema de Aprendizaje de Negociación Electrónica (SANE) como una propuesta para automatizar la negociación en los mercados electrónicos.

En una primera versión, se implementó un prototipo de mercado electrónico donde existe un agente software comprador, representante de un cliente y un agente software vendedor, representante de un tienda virtual, que interactúan usando un lenguaje simbólico para negociar el precio de un producto. Este mercado contempla todos los componentes que debería tener un proceso de negociación electrónica: ambiente o mercado, monitor, lenguaje de negociación, protocolo de negociación y participantes.

Como los buenos negociantes se hacen con el tiempo, se vio que era necesario representar los procesos de negociación de alguna forma que fuera útil en las siguientes oportunidades. Así: las reglas de decisión representan situaciones concretas, los árboles de decisión y las redes neuronales generalmente se usan para tareas de clasificación, los algoritmos evolutivos buscan obtener el valor óptimo, los algoritmos de refuerzo permiten saber si el agente ha actuado bien o no y los algoritmos de agrupamiento tienen como objetivo formar grupos. En cambio, el Razonamiento Basado en Casos permite representar cada proceso de negociación como un caso. Los agentes paulatinamente guardarán en su memoria los procesos de negociación realizados y los podrán aprovechar en las siguientes negociaciones.

En la segunda versión del proyecto, al agente software comprador se le incluyó Razonamiento Basado en Casos con el objetivo de que, en primer lugar, pueda aprender del usuario cómo negociar y en segundo lugar de que, en las siguientes oportunidades sea capaz de negociar por él. Así: cada vez que el agente comprador asiste a un proceso de

negociación en el mercado electrónico, puede adaptar éxitos pasados a su comportamiento actual. Se han implementado dos formas de adaptación de casos: El agente puede adaptar su comportamiento de un caso anterior en el que se haya comprado el producto requerido al menor precio o puede adaptarlo de un caso en que haya un comprador parecido a él. De los resultados obtenidos, se consiguen mejores precios al adaptar el comportamiento de un caso que consiga el menor precio. Asimismo, los resultados también muestran que los agentes que usan Razonamiento Basado en Casos consiguen mejores precios que los que no lo hacen. Finalmente, el comportamiento actual también se almacena en la base de casos de SANE, de manera que el agente comprador pueda disponer de mayor conocimiento con el transcurrir del tiempo. El comportamiento del agente se ha representado como una función matemática y por consiguiente, la adaptación de los casos se ha conseguido usando técnicas de regresión estadística.

A los usuarios de SANE, les agradó la idea de tener un agente software al que sólo se le proporcionan algunos parámetros de negociación y puede negociar en lugar de ellos en una tarea que se considera embarazosa y que consume bastante tiempo. Además, porque los humanos, en general, aceptan el primer precio que les parece razonable, en cambio, los agentes software son capaces de agotar las posibilidades y conseguir mejores precios, aunque en esta propuesta, no siempre se obtiene el precio óptimo. Sin embargo, los resultados alcanzados satisfacen las expectativas de los usuarios y como los precios a ofertar por el agente se calculan en tiempo real, la inserción de SANE en las actuales aplicaciones B2C sería factible.

Si bien se ha conseguido desarrollar agentes software compradores que aprenden de los usuarios y de otros agentes cómo regatear el precio de un producto en los mercados electrónicos, esta propuesta aún deberá evaluarse usando medidas económicas que permitan valorar objetivamente su desempeño. También, de acuerdo a la literatura estudiada no se han encontrado trabajos similares, por lo tanto es difícil comparar lo logrado con trabajos que sólo se parecen parcialmente. De igual forma, el comportamiento de los agentes aún es rústico y se ha pensado en refinarlo combinando el

Razonamiento Basado en Casos con otras técnicas del Aprendizaje Automático y áreas relacionadas.

Aunque aún resta un largo camino por seguir, los resultados obtenidos hasta el momento son alentadores y nos permiten pensar que se puede automatizar y personalizar el proceso de negociación automática en el comercio electrónico de manera importante en los próximos años.

6.2 Trabajo Futuro

El actual Sistema de Aprendizaje de Negociación Electrónica se piensa enriquecer con la realización de las siguientes tareas:

- Evaluar SANE con medidas de desempeño de negociación tales como: conjunto de beneficios/utilidad, balance de contrato, distancia a la frontera eficiente, distancia a la solución de regateo Nash, para conseguir resultados objetivos de su desempeño y de ser posible, compararlo con otros sistemas similares.
- Usar la memoria de casos negativos para que el agente también pueda aprender lo que no debe hacer en un proceso de negociación.
- Recuperar y adaptar no solamente uno, sino varios casos (positivos o negativos) para conseguir una solución que obtenga lo mejor de todos los casos recuperados.
- Incrementar el tamaño de la base de casos de SANE para la realización de los futuros experimentos.
- Refinar la interfaz de SANE para mejorar la interacción con los usuarios. Aunque el agente software comprador actualmente puede negociar por sí solo, se ha pensado en incorporar algunas ventanas de diálogo para que pueda consultarle al usuario diversas opciones y así, el sistema sería más interactivo y personalizado.
- Construir SANE en un ambiente Web.
- Combinar el Razonamiento Basado en Casos con otras técnicas de Aprendizaje Automático para conseguir comportamientos más sofisticados. Por ejemplo, se ha

pensado en añadir reglas como atributos de los casos o del algoritmo de negociación.

- Combinar el Razonamiento Basado en Casos con otras técnicas de Teoría de Juegos, Biología, Estadística, etc.

6.3 Divulgación de Resultados

Como resultado de la investigación, se ha presentado el siguiente artículo en un congreso nacional (ver Anexo A).

- C. Salazar-Serrudo, L. J. García-Villalba, “*Comercio Electrónico B2C Personalizado*”. Artículo aceptado en XXIII Simposium Nacional de la Unión Científica Internacional de Radio (URSI 2008), Facultad de Informática, Universidad Complutense de Madrid, 22 al 24 de septiembre de 2008, España .

Bibliografía

- [1] J. Brzostowski, R. Kowalczyk, “*On possibilistic Case-based Reasoning for Selecting Partners for Multi-attribute Agent Negotiation*”, Copyright 2005 ACM, pp. 273-279.
- [2] B. Berney, “*Software Agents – A review*”, Technical Report, Manchester Metropolitan University, 1999.
- [3] F. Breiman, S. Olshen, *Classification and Decision Trees*, Ed. Wadsworth, England, 1984.
- [4] D. Borrajo, “*Aprendizaje Automático*”, Curso de Doctorado en Inteligencia Artificial, Universidad Carlos III de Madrid, 2007. URL: <http://scalab.uc3m.es/~docweb/doctorado/aa/descripcion.html>
- [5] J. Carbonell, R. Michalski, T. Mitchell, *An overview of machine learning*, Ed. Springer Verlag, Berlin, 1984.
- [6] A. C. Chapman, A. Rogers and N. R. Jennings, “*Learn While You Earn: Two Approaches to Learning Auction Parameters in Take-it-or-leave-it Auctions*”, Proc. Of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008), Padgham, Parkes, Muller and Parsons (eds.), May 12-16, Estoril, Portugal.
- [7] A. Chávez, P. Maes, “*Kasbah: An agent marketplace for buying and selling goods*”, Proc. Application of Intelligent Agents and MultiAgent, London, UK, 1996, pp. 75-90.
- [8] J. Chen, K. Chao, N. Godwin, C. Reeves, P. Smith, “*An automated negotiation mechanism based on co-evolution and game theory*”, SAC 2002, Madrid, España, Copyright 2002 ACM.
- [9] K. Crowston, “*The effects of market-enabling Internet agents on competition and prices*”, *Proceedings of the Seventeenth International Conference on Information Systems*, Cleveland, OH, December 1996, pp.381-390.

- [10] P. Dasgupta, N. Narasimhan, L.E. Moser and P.M. Melliar-Smith, “*MAGNET: Mobile Agents for Networked Electronic Trading*”, Technical Report, University of California, CA, 1998, USA.
- [11] C. Duarte, B. Borsoi, “*Negociacio de preco em comércio electronico: uma investigacao experimental*”, Brasil, 2003.
- [12] S. Fatima, M. J. Wooldridge and N. R. Jennings, “*Optimal Negotiation Strategies for Agents with Incomplete Information*”, Springer Verlag, ATAL 2001, pp. 53-68.
- [13] S. Fatima, M. J. Wooldridge and N. R. Jennings, “*A comparative Study of Game Theoretic and Evolutionary Models of Bargaining for Software Agents*”, Artificial Intelligence Review, Vol. 23, pp. 185-203, 2005.
- [14] G. Fernández, “*Aprendizaje en Máquinas*”, Apuntes del curso de Aprendizaje en Máquinas del Máster en Telecomunicaciones, Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid, España, 1998.
- [15] K. Goh, H. Teo, H. Wu, K. Wei, “*Computer-supported negotiation: an experimental study of bargaining in electronic commerce*”, ACM Web, November, 2007.
- [16] R. Guttman, A. Moukas, P. Maes, “*Agents as mediators in electronic commerce*”, Electronic Markets, pp. 22-27, vol. 8, No. 1, 1999.
- [17] M. He, N.R. Jennings, H.F. Leung, “*On Agent-Mediated Electronic Commerce*”, IEEE Transactions on Knowledge and Data Engineering, Vol. 15, No. 4, July/August 2003.
- [18] S. Huang, F. Lin, “*Designing Intelligent Sales-agent for Online Selling*”, ICEC’05, August 15-17, 2005, X’ian, China, Copyright 2005 ACM.
- [19] Instituto Nacional de Estadística, “*Encuesta sobre el uso de TICs y Comercio Electrónico en las empresas 2006/07*”, 2007, España.
- [20] K. Jonkheer, “*Intelligent Agents, Markets and Competition: Consumers*”, Interests and Functionality of Destination Sites”, URL: http://www.firstmonday.dk/issues/issue4_6/jonkheer/index.html, 2006.
- [21] B. Krulwich, “*Automating the Internet: Agents as User Surrogates*”, Technical Report for Agentsoft Ltd., Jerusalem, Israel, 1997.

- [22] M. Kubat, I. Bratko, R. Michalski, *Machine Learning and Data Mining: Methods and Applications*, John Wiley & Sons Ltd., 1996.
- [23] R. Lau, “*Adaptive Negotiation Agents for E-business*”, ICEC’05, August 2005, Copyright 2005 ACM.
- [24] A. Lomuscio, M. Wooldridge, N. R. Jennings, “*A Classification Scheme for Negotiation in Electronic Commerce*”, *Lecture Notes in Computer Science*, Ed. Springer – Verlag, Vol. 1991, pp. 19-33, 2001.
- [25] P. Maes. Página de Pattie Maes. URL: <http://web.media.mit.edu/~pattie/>, Agosto 2008.
- [26] P. Maes, "Pattie Maes on Software Agents: Humanizing The Global Computer", *IEEE Internet Computing*, vol. 01, no. 4, pp. 10-19, July-August, 1997, <http://doi.ieeecomputersociety.org/10.1109/MIC.1997.10034>
- [27] P. Maes, “*Software Agents*”, Technical Report, Software Agents Group, MIT Lab, 2002.
- [28] N. J. Nilson, “*Introduction to Machine Learning: An early draft of a proposed textbook*”, Technical Report, Robotics Laboratory, Department of Computer Science, Stanford University, Stanford, USA, 1996.
- [29] N. J. Nilsson, *Inteligencia artificial: Una nueva síntesis*, Ed. McGraw-Hill Interamericana, España, 2001.
- [30] F. Peyman, N. R. Jennings, A. R. Lomusci, S. Parsons, C. Sierra & M. Wooldridge, “*Automated Negotiation: Prospects, Methods and Challenges*”, *Int. J. of Group Decision and Negotiation*, Vol. 10, No 2, pp. 199-215, 2001.
- [31] E. Plaza, A. Aamodt, “*Case-Based Reasoning, Foundational Issues, Methodological Variations and System Approaches*”, *Artificial Intelligence Communications*, Vol. 7, No. 1, pp. 39-59, 1994.
- [32] J. R. Quinlan, *C4.5 Programs for Machine Learning*, Ed. Morgan Kauffman, 1993.
- [33] J. R. Quinlan, “*Learning Logical Definitions from Relations*”, *Machine Learning*, Vol. 5, pp. 239-266, 1990.
- [34] J. R. Quinlan, “*Knowledge acquisition from structured data*”, *IEEE EXPERT*, Vol. 6, No 16, pp. 32-37, 1991.

- [35] J. R. Quinlan, M. Cameron-Jones, “*FOIL: A midterm report*”, Basser Department of Computer Science, University of Sydney, Australia, 1993.
- [36] I. Rahwan, R. Kowalczyk, H. Pham, “*Intelligent Agents for Automated One-to-Many e-Commerce Negotiation*”, Twenty-Fifth Australian Computer Science Conference (ACSC2002), Melbourne, Australia.
- [37] I. Rahwan, L. Sonenberg, N. Jennings, P. McBurney, “*STRATUM: A methodology for designing heuristic agent negotiation strategies*”, Applied Artificial Intelligence, 21:489-527. Copyright 2007 Taylor & Francis Group.
- [38] J. A. Recio-García, B. Díaz-Agudo, “*An introductory user guide to JCOLIBRI 0.3*”, Technical Report No. 144/2004, Depto. Sistemas Informáticos y Programación, Universidad Complutense de Madrid, Spain, November 2004.
- [39] A. Ribeiro, “*Algoritmos de aprendizaje aplicados a minería de datos*”, Texto de apoyo Curso de Doctorado Minería de Datos, Departamento de Sistemas y Lenguajes de Programación de la Facultad de Informática de la Universidad Politécnica de Madrid, 1998.
- [40] E. Rich, K. Knight, *Inteligencia Artificial*, Ediciones McGraw-Hill Interamericana de España, España, 1994.
- [41] S. Russell, P. Norvig, *Inteligencia artificial: Un enfoque moderno*, Ed. Prentice Hall Hispanoamericana, S.A., México, 1996.
- [42] C. Salazar, C. Delrieux, “*Aprendizaje Basado en Casos aplicado a Procesos de Negociación*”, IX Workshop de Investigadores en Ciencias de la Computación (WICC’07), 3 y 4 de mayo de 2007, Trelew, Argentina.
- [43] C. Salazar, J. García-Villalba, “*Aprendizaje Interactivo e Incremental en Procesos de Negociación en Mercados Electrónicos*”, 3er Workshop International sobre Aplicaciones Prácticas de Agentes y Sistemas Multiagentes (IWPAAM’S 2004), 13–15 Octubre 2004, Burgos, España.
- [44] T. Sandholm, “*Automated Negotiation*”, Communication of the ACM, Vol. 42, No. 3, March 1999.
- [45] T. Sandholm, “*An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations*”, Proceedings of the Eleventh National Conference on Artificial Intelligence, pp. 256-262, 1993

- [46] C. Serrano Cinca, “*El Comercio Electrónico en los departamentos de una empresa*” [en línea] 5campus.org, Sistemas Informativos Contables, <http://www.5campus.org/leccion/econta>, Octubre 2007.
- [47] C. Sierra, “*Agent-Mediated Electronic Commerce*”, *Autonomous Agents and Multi-Agent Systems*, 9, 285-301, 2004, Kluwer Academic Publishers, The Netherlands, 2004.
- [48] M. Tsvetovatyy, M. Gini, B. Mobasher and Z. Wieckowski, “MAGMA: An agent-based Virtual Market for Electronic Commerce”, *International Journal of Applied Artificial Intelligence*, (11: 6), September 1997, pp. 501-523.
- [49] I. Watson & F. Marir, “*Case-Based Reasoning: A Review*”, *The Knowledge Engineering Review*, Vol. 9 No. 4, 1994.
- [50] S. Zhang, F. Makedon, “*Privacy Preserving Learning in Negotiation*”, 2005 ACM Symposium on Applied Computing, USA.
- [51] S. Zhaohao, R. Gavin, “*Intelligent Techniques in E-commerce: A case Based Reasoning Perspective: Studies in Fuzziness and Soft Computing*”, Publisher Springer – Verlag Berlin Heidelberg 2004, Germany.

Anexo A

Artículo: “Comercio Electrónico B2C Personalizado”