



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

Departamento de Ingeniería Eléctrica

Sección de Computación

**Diseño y construcción de un portal de agentes de
compra para comercio electrónico**

Tesis que presenta

Paola Elizabeth Ramírez Santiago

para obtener el Grado de

Maestra en Ciencias

en la Especialidad de

Ingeniería Eléctrica

Director de la Tesis

Dr. José Oscar Olmedo Aguirre

México, D.F.

Septiembre 2004

Resumen

El comercio electrónico es una tecnología que permite a consumidores y proveedores de bienes y servicios tener acceso a información comercial y realizar transacciones en forma sencilla y económica. Los consumidores finales pueden consultar fácilmente los productos de varias compañías virtualmente desde cualquier lugar y así tomar las mejores decisiones en menos tiempo.

En la actualidad, existen portales en Internet dedicados al comercio electrónico que requieren la intervención directa del comprador, el cual realiza la búsqueda, selección, compra y pago de productos y servicios. Desafortunadamente, esta forma de conducir los negocios no elimina o reduce compras rutinarias, además de ello no delega la actividad de compra a un agente, logrando que el comprador invierta grandes cantidades de tiempo en realizar esta tarea.

En este trabajo se describe y construye un prototipo de un modelo de tienda para comercio electrónico del tipo B2C y B2B denominado *STD*. El sistema consiste de un portal Web que se encarga de satisfacer listas de pedidos mediante agentes de software. Los agentes son los facultados para realizar la búsqueda del pedido en las tiendas de preferencia del cliente. El usuario contratará el agente visitando el portal STD en Internet para darle a conocer la lista de productos que desea adquirir así como las tiendas que debe visitar siguiendo el orden de su preferencia. Al final de su recorrido el agente notifica al usuario cuáles son los productos que adquirió, cuáles no, en cuáles tiendas y a cuál precio.

La contribución del trabajo consiste en el desarrollo de un modelo de tienda que implementa un agente de búsqueda basado en servicios Web. El agente permite buscar y comprar listas de productos en diversas tiendas usando varios criterios de selección. Con el modelo se busca ayudar a resolver algunos de los problemas no atendidos por los portales dedicados a compradores. Además de ello, el modelo permite acceder a la funcionalidad de un sistema intermediario de comercio electrónico. La importancia de la solución puede apreciarse en el aumento de la actividad comercial en tiendas (tradicionales) y tiendas en línea, así como el beneficio social que conlleva.

Abstract

Electronic commerce is an information technology that allows consumers and providers of goods and services to get access to commercial information and to undertake transactions in a simple and an economic manner. The final consumers can consult easily the products offered by business located virtually at any place in the world and then to adopt the best decisions in less time.

Nowadays, there are portals on Internet dedicated to electronic commerce that need the direct intervention of the buyer who conducts the search, selection and payment of products and services. Unfortunately, this form of carrying on commercial businesses does not reduce routine buys. Besides, current forms of electronic commerce does not allow to the users to delegate the shopping related activities to any form of autonomous entities, leading to an unacceptable waste of time on performing these tasks.

This work describes and constructs a prototype of a shopping model for B2C and B2B electronic commerce called STD. The system consists of a Web portal designed to fulfill orders of lists of products by means of software agents. The agents are authorized to search the products in the shops according to the preferences of their clients. The user can hire a shopping agent by visiting the STD portal on Internet to indicate the product list that he or she wants to acquire along with the shops that the agent must visit following the order given by the user. At the end of its journey, the agent notifies to its user which are the products it found, in what shops were found and the selling price of those products. The agent also informs about the products that were not found in any shop.

The contribution of this work consists on developing a shopping model that implements a search & buy software agent based on Web services. The agent allows the user to buy product lists that can be searched on diverse shops following several criteria of selection such as minimum price or least delivery time. With this shopping model we hope to help to solve some of the problems not attended yet by Web portals for electronic commerce. In addition, the model provides the means to get access to the functionality of a brokering system of B2B electronic commerce. The importance of the proposed solution can be appreciated in the increase of the commercial activity observed on online shops along with the social benefits that this bears.

Agradecimientos

Índice general

Resumen	iii
Abstract	v
Agradecimientos	vii
1. Introducción	1
1.1. El comercio electrónico	2
1.2. Agentes de comercio electrónico	4
1.3. Planteamiento del problema	5
1.4. Objetivos	6
1.5. Metodología	7
1.6. Contribuciones	8
1.7. Organización del documento	9
2. Comercio electrónico basado en agentes	11
2.1. Introducción	11
2.2. Agentes de software	12
2.3. Comercio electrónico con agentes	13
2.4. Tipos de agentes	15
2.5. Otros agentes	19
2.6. Servicios Web	20
2.7. HTML	21
2.8. XML	22
2.9. SOAP	25
2.10. WSDL	26
2.11. UDDI	27
2.12. Resumen	28
3. Caso de estudio: Búsqueda de productos en STD	29
3.1. Consideraciones iniciales	29
3.2. Búsqueda de productos en STD	30
3.2.1. Bienvenida al sistema	31
3.2.2. Registro de usuarios	32
3.2.3. Navegando por los productos	33
3.2.4. Pedido	33
3.2.5. Búsqueda	36
3.2.6. Resultados	37
3.2.7. Carrito de compra	38
3.2.8. Ayuda	39

3.3. Resumen	40
4. Diseño del sistema STD	41
4.1. Descripción organizacional	41
4.2. Descripción funcional	44
4.2.1. Funcionalidad de STD	44
4.2.2. Funcionalidad de los servidores comerciales	48
4.3. Descripción de los datos	50
4.3.1. Base de datos principal de STD	50
4.3.2. Bases de datos de los servidores comerciales	53
4.4. Descripción del comportamiento	55
4.5. Resumen	61
5. Implementación del sistema STD	63
5.1. Descripción organizacional	63
5.2. Descripción funcional	64
5.2.1. Descripción de las clases de STD	66
5.2.2. Descripción de las clases de los servidores comerciales	76
5.3. Descripción de los datos	81
5.4. Descripción del comportamiento	81
5.5. Resumen	87
6. Conclusiones y trabajo a futuro	89
6.1. Conclusiones	89
6.2. Trabajo a futuro	90
Bibliografía	94

Índice de figuras

2.1. Registro e invocación de un servicio Web	20
2.2. Ejemplo de una DTD	23
2.3. Estructura de un mensaje SOAP	25
2.4. Funcionamiento general de UDDI	27
3.1. Caso de estudio: Búsqueda de productos en STD	31
3.2. Pantalla inicial del sistema	32
3.3. Registro del cliente	32
3.4. Listas de productos	33
3.5. Elaboración y visualización del pedido	34
3.6. Tiendas registradas	34
3.7. Selección de tiendas	34
3.8. Envío de solicitudes y resultados entre STD y las tiendas	36
3.9. Resultados de la búsqueda	37
3.10. Resultados de la búsqueda y productos agregados al carrito	38
3.11. Carrito de compra	38
3.12. Ayuda del sistema	39
4.1. Diseño arquitectural de STD	41
4.2. Interacción cliente – servidor Web	43
4.3. Tablas de las base de datos de STD y sus relaciones	51
4.4. Tablas de las base de datos de las empresas A y B	53
4.5. Tablas de Pedidos de las empresas A y B	54
4.6. Tabla de la base de datos de la empresa GiganteSTD	55
4.7. Diagrama de Casos de uso de STD	56
4.8. Diagrama de secuencia del registro de clientes	56
4.9. Diagrama de secuencia de administración de listas de clientes	56
4.10. Diagrama de secuencia de registro de tiendas	57
4.11. Diagrama de secuencia de actualización de datos de tienda	58
4.12. Diagrama de secuencia de administración de listas de productos	58
4.13. Diagrama de secuencia de administrar pedido	59
4.14. Diagrama de secuencia de búsqueda de productos	59
4.15. Diagrama de secuencia de administración del carrito de compras	60
4.16. Diagrama de secuencia de búsqueda de productos en servidores comerciales	61
5.1. Clases de STD y de los servidores comerciales	64
5.2. Mensaje de invocación al servicio Web <i>getProductDetails</i> desde servidor comercial <i>Empresa A</i>	80
5.3. Mensaje de respuesta generado desde el servidor comercial <i>Empresa A</i> al invocar su servicio Web <i>getProductDetails</i>	80

Índice de cuadros

5.1. Clase Registro	66
5.2. Clase Tiendas	67
5.3. Clase Clientes	68
5.4. Clase Productos	68
5.5. Clase Pedidos	69
5.6. Clase Construye_pedido	69
5.7. Clase Mostrar_pedido	70
5.8. Clase Armar_pedido	70
5.9. Clase Resultado	71
5.10. Clase Mostrar_resultado	72
5.11. Clase Resultado_carrito	72
5.12. Clase Armar_resultado	72
5.13. Clase Seleccionar_tiendas	73
5.14. Clase Mostrar_tiendas	73
5.15. Clase Armar_seleccionar_tiendas	74
5.16. Clase CreaHilo	74
5.17. Clase InvocaHilos	75
5.18. Clase WSDLManipulation	75
5.19. Clase Carrito	76
5.20. Clase AmazonSearch	79

Capítulo 1

Introducción

Con la llegada de Internet se ha creado un nuevo concepto en el campo de la economía donde existe la posibilidad de comprar y vender productos. En la práctica las empresas están comenzando a usar Internet como un nuevo canal de ventas, sustituyendo las visitas personales y el teléfono por pedidos electrónicos, ya que realizar un pedido por Internet cuesta menos que hacerlo por vías tradicionales. Nace entonces el comercio electrónico como una alternativa de reducción de costos y una herramienta fundamental en el desempeño empresarial.

El comercio electrónico puede definirse como el intercambio de bienes y servicios a través de Internet. El comercio electrónico es una tecnología que permite tener acceso a información comercial y realizar transacciones electrónicas en forma sencilla y económica. Es la venta a distancia aprovechando las grandes ventajas que proporcionan las nuevas tecnologías de la información, como la ampliación de la oferta, la interactividad y la inmediatez de la compra. El comercio electrónico hace posible adquirir y ofrecer productos y servicios de manera eficiente a través de redes privadas o de Internet.

Las compañías que se dedican a la venta de productos y servicios pueden ofrecer estos automatizando todo el proceso de compra y pueden también analizar los patrones de compra de sus clientes con el propósito de ofrecerles un mejor servicio y, potencialmente, incrementar sus ventas.

En este capítulo se muestra un panorama general del comercio electrónico, se habla sobre los tipos, modelos y características del comercio electrónico, asimismo se define a un agente de software. Al final del capítulo se describen los objetivos, la propuesta y las contribuciones de este trabajo de investigación que pretende resolver algunos de los problemas encontrados dentro del comercio electrónico.

1.1. El comercio electrónico

Dentro del comercio electrónico existen diferentes formas en las que se pueden establecer negocios. Estas formas se centran en las personas o entidades que intervienen en la relación comercial, cuyas interacciones se pueden establecer de diferentes formas. Debido a ello podemos definir diferentes tipos de comercio electrónico:

Negocio – Negocio. Conocido como B2B (*Business to business*). Este tipo de comercio electrónico establece vínculos de comercio entre empresas. Es decir, una empresa hace uso de una red para formular órdenes de compra a sus proveedores (otras empresas), recibir facturas y realiza pagos correspondientes.

Negocio – Cliente. Llamado comúnmente B2C (*Business to customer*). Dentro de esta categoría, las empresas mantienen una relación directa con el consumidor. Las empresas establecen en Internet una "tienda virtual" en la que promocionan sus productos y/o servicios, los clientes podrán acceder a las tiendas y comprar los productos de su preferencia.

Cliente – Cliente. Denominado C2C (*Customer to customer*). Este tipo de comercio electrónico tiene lugar cuando individuos establecen relaciones comerciales entre sí, aprovechando los servicios que ofrecen algunos sitios como son las subastas.

Cliente – Negocio. Conocido como C2B (*Customer to business*). Se establece cuando un grupo de consumidores se unen para adquirir mayor fuerza y establecer pedidos con empresas.

Negocio – Administración. Mejor conocida como A2B (*Administration to business*). Se refiere a vínculos comerciales que establecen entre empresas y las diferentes organizaciones de gobierno.

Empresa – Trabajador. Nombrado como B2E (*Business to employee*). Intercambio comercial que se lleva a cabo entre trabajadores de una empresa y esta.

Amigo – Amigo. Denominado P2P (*Peer to peer*). Este tipo de comercio electrónico, se establece cuando mediante el uso de ciertos programas los individuos a quienes se les denomina con el nombre de *amigos* realizan algún tipo de intercambio según sus intereses. p.ej. El intercambio de música.

En el mundo de Internet se encuentran una gran variedad de portales de comercio electrónico que implementan algún modelo de comercio electrónico. Un modelo de comercio electrónico es una forma de organizar y conducir las transacciones comerciales. La elección del tipo de modelo de comercio electrónico que utiliza cada empresa depende

principalmente de las políticas, operaciones, tecnología e ideología que maneje. Existen varios modelos de negocios que actualmente se pueden encontrar en Internet entre los cuales podemos mencionar los siguientes:

- Modelo de la tienda frontal (*Storefront model*)
- Modelo de portal (*Portal model*)
- Modelo de subasta (*Auction model*)
- Modelo de nombra tu precio (*Name-Your-Price model*)
- Modelo de comparación de precios (*Comparison Pricing model*)
- Modelo de demanda sensitiva (*Demand Sensitive Pricing model*)

Modelo de la tienda frontal

El modelo de tienda frontal permite a los comerciantes vender productos en la Web. En este tipo de modelo se define el concepto de lo que conocemos como *tiendita* sólo que ahora podemos tenerla en la casa mediante el uso de una computadora. Una de las principales ventajas que reflejará una empresa al utilizar este modelo, es que el negocio estará disponible para consumidores de cualquier parte del mundo las 24 horas del día durante todos los días del año lo que a diferencia de las tiendas comunes no se podría lograr.

- Catálogo de productos en línea
- Orden de procesamiento
- Seguridad en el pago
- Cumplimiento del tiempo de entrega del producto

En el modelo de tienda frontal se encuentra: la tecnología del carrito de compra y los centros comerciales en línea.

Tecnología del carrito de compra

Esta tecnología permite que en un portal se establezca una opción en la que se permita a los clientes acumular una lista de los artículos que desean comprar, dándoles la posibilidad de modificar su decisión de compra en el momento en que así lo deseen tal y como sucede en un supermercado.

El carrito de compra contiene un catálogo de productos donde se almacenan todos los productos en existencia, también se debe implementar la tecnología de las bases de datos ya que con ello es más sencillo el manejo de los productos.

Centros comerciales en línea

Un centro comercial en línea es una colección de proveedores que ofrecen sus productos dentro de un mismo sitio. Este concepto se tomó de los centros comerciales tradicionales donde se puede encontrar muchos productos diferentes en un solo lugar, sin tener que visitar varios lugares para obtener lo que se desea. En los centros comerciales en línea se maneja un concepto semejante, los clientes no buscan en varios sitios los productos que necesitan sino que un centro comercial en línea ofrece la posibilidad de encontrar una gran diversidad de productos.

Los centros comerciales en línea actúan como portales de compra, dirigen la búsqueda de un producto hacia un lugar en específico donde el cliente lo pueda encontrar. Ofrecen también velocidad en cuanto a la búsqueda de productos.

1.2. Agentes de comercio electrónico

El comercio electrónico brinda la posibilidad de emplear agentes con la finalidad de automatizar muchas de las tareas que desarrolla. Un agente de software es un término utilizado de forma general para definir un programa que actúa en representación de un usuario, auxiliando en la ejecución en alguna tarea. En la filosofía de programación por agentes, un usuario delega tareas al agente para que las ejecute en su nombre.

Para poder actuar como un agente de software, un objeto debe ser capaz de ejecutarse en un ambiente específico y poseer las siguientes características esenciales:

Autonomía: Ejerce el control sobre sus propias acciones

Permanencia: Un agente tiene la capacidad de subsistir ó permanecer aún después de terminadas la acciones que le fueron solicitadas

Adaptabilidad: Es capaz de adaptarse al entorno donde se encuentre

Además de estas características, los agentes pueden tener otras más. Un agente tiene la capacidad de comunicarse ya sea con otros agentes o con personas con el fin de conseguir u obtener la información que necesita, presentando así la propiedad de **comunicación**. Estableciendo de esta manera un mecanismo de comunicación uniforme que le permite intercambiar información con otros agentes. Un agente puede presentar

la característica de **aprendizaje**, siendo capaz de aprender en base a sus experiencias o situaciones vividas.

La **movilidad** es otra característica que puede poseer un agente de software. Los agentes que presentan esta característica son comúnmente conocidos como agentes móviles. Un agente móvil debe ser capaz de interrumpir su ejecución para transportarse de un lugar a otro y retomarla, estableciendo así un paradigma de programación entre objetos distribuidos.

1.3. Planteamiento del problema

En la actualidad, podemos encontrar una amplia variedad de portales dedicados al comercio electrónico entre algunos de los cuales podemos mencionar *Bargain Finder* (<http://bf.cstar.ac.com>), *FireFly* (<http://www.firefly.com>) o *Kasbah* [1], basados en la implementación de algún modelo de comercio electrónico. En estos portales los usuarios de servicios y compradores de productos pueden buscar, seleccionar, comprar y pagar por medios electrónicos.

Desafortunadamente, esta forma de conducir el comercio electrónico requiere la intervención directa del usuario, lo cual impide el desarrollo de otras posibilidades más convenientes tanto para compradores como para comercios.

Entre las dificultades que presentan los portales orientados a compradores podemos señalar las siguientes:

1. **Actividades rutinarias.** Aún cuando los portales en Internet han facilitado y aumentado enormemente la actividad comercial, existen actividades rutinarias para los que este enfoque es aún inapropiado. Por ejemplo, la compra de productos de uso cotidiano como comestibles, productos de limpieza y perfumería en donde los compradores consistentemente adquieren el mismo tipo, marca y presentación del producto son actividades que podrían automatizarse. El tiempo dedicado a la compra de productos de uso común podría disminuir significativamente si los sitios comerciales ofrecieran facilidades que permitieran a los compradores recordar la lista de productos que generalmente adquieren.
2. **Alcance limitado.** A pesar de la promesa del comercio electrónico de poder realizar transacciones comerciales desde cualquier lugar y en cualquier momento, la experiencia ha demostrado que los compradores requieren facilidades adicionales que les permitan contratar agentes de software para comprar bienes o contratar

servicios cuando los compradores a quienes representan no dispongan del tiempo necesario o posean discapacidades físicas que se los impidan.

3. **Falta de aprovechamiento de recursos.** En la actualidad, un gran número de comercios poseen la infraestructura necesaria para la entrega de productos a domicilio. Con un servicio de entrega eficaz y confiable, los comercios proporcionan un beneficio considerable a la sociedad no solo para llevar productos indispensables para personas discapacitadas sino también contribuyen a reducir el tráfico vehicular en ciudades con elevados índices de congestión.
4. **Falta de interoperabilidad.** En el comercio electrónico no se le ha concedido suficiente importancia a los beneficios que representaría para sus clientes la ayuda en la preparación de declaraciones para la recaudación fiscal. En productos o bienes de consumo con carga impositiva, los agentes de software podrían enviar esta información a portales de contaduría pública en donde se colectaría y prepararía la información fiscal correspondiente. Para que esto sea posible, se requiere que los sistemas de comercio electrónico sean interoperables con sistemas externos.

En este trabajo de investigación se buscó resolver específicamente los problemas descritos en los puntos 1 y 2 antes mencionados, definiendo un sistema de comercio electrónico B2C y B2B el cual fue nombrado como *Surte tu Despensa* y al que de ahora en adelante se identifica con las siglas *STD*. *STD* permitire definir y enviar agentes que realizan compras rutinarias. Los agentes son quienes se encargan de satisfacer las listas de pedidos mediante su búsqueda en las diferentes tiendas de preferencia, finalizando con los resultados obtenidos en cada una de las mismas.

1.4. Objetivos

El objetivo de este trabajo consiste en definir un modelo simple de agente de compra y construir un prototipo experimental que permita contratar agentes que realicen compras mediante listas de productos.

El sistema consiste de un portal Web que se encarga de satisfacer listas de pedidos mediante agentes de software. Los agentes son los facultados para realizar la búsqueda del pedido en las tiendas de preferencia del cliente. El usuario contrata al agente visitando un portal en Internet para darle a conocer la lista de productos que desea adquirir, así como las tiendas que debe visitar siguiendo el orden de su preferencia. Al final de su

recorrido el agente notifica al usuario cuáles son los productos que ha adquirido, cuáles no, en qué tiendas y a que precio.

Para realizar este objetivo se realizaron los siguientes objetivos particulares:

- Diseñar y construir un portal Web que permita definir y satisfacer listas de pedidos
- Definir agentes de software que tendrán como propósito realizar las búsquedas de pedidos
- Diseñar y construir el modelo de datos del sistema
- Implementar el sistema STD mediante el desarrollo de servicios Web

1.5. Metodología

Los pasos que se siguieron para el desarrollo del sistema fueron los siguientes:

- Definir la organización general de los servidores de comercio electrónico y del portal de agentes en base a la arquitectura cliente – servidor de la Web.
- Definir el protocolo de comunicación que usarán los agentes para solicitar, buscar, seleccionar, devolver, aceptar, rechazar productos o servicios.
- Definir las reglas que regulan la actividad comercial para cada una de las operaciones del protocolo de comunicación.
- Diseñar y construir el portal que corresponde a la central de agentes de compra. En este portal, se contratan e instruyen a los agentes sobre la lista de productos que deben adquirir.
- Diseñar y construir un servidor que atienda las peticiones de los agentes de compra de acuerdo al protocolo de comunicación aceptado y siguiendo las reglas del comercio.

1.6. Contribuciones

Las contribuciones obtenidas al finalizar este trabajo fueron las siguientes:

- **Un modelo de tienda.** Este trabajo implementa un modelo de tienda virtual. El modelo permite el registro de diversas empresas proveedoras de productos. Además, el modelo de tienda virtual ofrece a los usuarios un catálogo de productos. Los productos contenidos en el catálogo están integrados conforme a los existentes en la diversas tiendas registradas en el sistema. El modelo de tienda utiliza un agente para desarrollar la búsqueda de los productos. Tanto el agente de búsqueda como el modelo de tienda, basan la búsqueda de los productos utilizando la tecnología de servicios Web. Para que una empresa pueda registrarse debe adoptar modelo de tienda del sistema, esto se logra mediante el desarrollo de los servicios Web donde ofrecerá sus productos. Además deberá adaptarse al estándar de categorización de productos desarrollado en este trabajo.

- **Un portal Web.** La interfaz sobre la que el usuario solicita las búsquedas consiste en un portal Web. El portal Web trabaja bajo la tecnología JSP. El portal cuenta con una estructura de tienda en línea. El portal Web permite acceder a la funcionalidad de un sistema intermediario de comercio electrónico, logrando con ello acrecentar la búsqueda de productos hacia las empresas que se encuentran registradas en el sistema intermediario.

- **Un agente de búsqueda.** El programa encargado de desarrollar la búsqueda consiste en un *agente de búsqueda*. El agente se encarga de buscar y comprar las listas de productos elaboradas por los clientes dentro del portal Web. Para el logro de la búsqueda, el agente recorre cada una de las tiendas de preferencia del cliente solicitándoles los productos de la lista. Las tiendas de preferencia del cliente deberán estar registradas dentro del modelo de tienda antes de iniciar con la búsqueda. Como resultado de la búsqueda, el agente presenta al usuario un listado de los productos que encontró en las tiendas de su preferencia. El usuario podrá seleccionar del resultado de la búsqueda, los productos que mejor cumplan con sus intereses finalizando con ello su compra.

1.7. Organización del documento

El documento de tesis está organizado de la siguiente manera. En el capítulo 2 se introducen los conceptos fundamentales para el buen entendimiento de la tesis como son agentes de software, portales y servicios Web. El capítulo 3 describe el funcionamiento general del sistema, se analiza un caso de estudio que tiene por objeto probar la funcionalidad del sistema desarrollado. En el capítulo 4 se describe el diseño del sistema enfocado desde cuatro aspectos diferentes: organización, de datos, funcional y de comportamiento. En el capítulo 5 se desarrolla la interfaz que corresponde a la implementación del sistema. Finalizaremos este trabajo de tesis en el capítulo 6 discutiendo las conclusiones y el trabajo a futuro.

Capítulo 2

Comercio electrónico basado en agentes

El objetivo de este capítulo es dar un panorama sobre los agentes de software definiendo los conceptos más importantes. Se especifican las principales ventajas del uso de agentes dentro del comercio electrónico. Se describen los tipos de agentes y cómo han sido implementados en algunas aplicaciones Web. Algunas de las aplicaciones Web que han tenido gran éxito al utilizar agentes, se describen de manera general en este capítulo. Al final del capítulo se explica la tecnología utilizada en el desarrollo de agentes dentro de este trabajo de investigación: servicios Web. Se puntualiza el concepto de servicios Web así como su funcionamiento general y se explican algunas de las tecnologías involucradas en el desarrollo de servicios Web como son XML, SOAP, WSDL y UDDI.

2.1. Introducción

El comercio electrónico engloba todas las características contenidas en el comercio electrónico tradicional entre las cuales podemos mencionar ofertas y promociones, negociación, entrega de productos, reclamaciones, etc. Además de desarrollar estas características adquiere otras más por el hecho de desarrollarse por medios electrónicos. Finalmente, podemos señalar que gracias a los contactos mediante las redes digitales es posible estar presente en un punto de venta, una transacción o en el lugar de un cliente remoto.

2.2. Agentes de software

Un agente de software puede definirse como un sistema informático que está situado en cierto entorno y que tiene la capacidad de actuar autónomamente en ese entorno normalmente en representación de alguien satisfaciendo sus objetivos de diseño. Podríamos asemejar su comportamiento al de un agente humano.

Como se describió en la sección 1.2 los agentes de software se diferencian del software tradicional porque poseen las siguientes características: autonomía, permanencia, adaptabilidad, comunicación, aprendizaje y movilidad, mismas que los hacen útiles en una amplia variedad de información y en la administración de procesos [2].

De acuerdo con la visión de agentes, los sistemas complejos de software serán reemplazados por éstos. Los sistemas de software cada día se vuelven más complejos debido a la demanda en el manejo de grandes cantidades de información. Esto provoca que los sistemas de software se vuelvan difíciles de manipular. Grandes empresas como *Microsoft* (<http://www.microsoft.com>), construyen sistemas compuestos por millones de líneas de código, lo que provoca que se vuelvan propensos a aumentar la cantidad de errores. Por esta razón, la tendencia se vislumbra en descomponer los sistemas en pequeños componentes independientes unos de otros. Los agentes pueden construirse de forma independiente con la condición de que posean un lenguaje de comunicación estándar que los ayude a comunicarse unos con otros. Además de ello, pueden ser integrados, modificados o eliminados de un sistema aunque este se encuentre en ejecución.

La tecnología de agentes está cambiando la forma tradicional de estructurar el software. El software tradicional se encuentra construido en capas, lo que requiere que cada una de ellas sea diseñada para conformar una sola interfaz [3]. De esta forma encontramos que es difícil combinar programas escritos en diferentes lenguajes, y se torna imposible integrar programas escritos para diferentes tipos de sistemas operativos. Los agentes son programas independientes unos de otros. Cada agente puede ser escrito en un lenguaje de programación diferente. Los agentes requieren cierta estandarización sólo a nivel de comunicación, estableciendo de esta forma un lenguaje común de comunicación entre ellos. Debido a que cada agente es independiente de otro, cada uno es construido cumpliendo al máximo sus objetivos de diseño, aumentando de esta forma la calidad en el desarrollo del mismo y del sistema.

Los agentes establecen sus orígenes en cuatro áreas de investigación: la robótica, la inteligencia artificial, los sistemas distribuidos y la computación gráfica. Las investigaciones sobre agentes en la robótica y la inteligencia artificial se encuentran relacionadas.

En la robótica, robots como SHAKEY [4], fueron programados para exhibir comportamientos autónomos en ambientes predefinidos. El primer agente de software probablemente el denominado ELIZA [5], programa que podía establecer una conversación con un usuario. Otro agente similar fue SHRLUD [6] que permitía establecer una comunicación con un robot virtual. Marvin Minsky en su trabajo "Society of Mind" [7] basó sus investigaciones en la mente humana como un sistema complejo compuesto por una colección de agentes, cada uno de los cuales se especializaba en un determinado dominio.

Las investigaciones en las áreas de sistemas distribuidos centraron su atención en la cooperación y comunicación entre agentes. Basaron sus objetivos en construir agentes con capacidades y conocimientos limitados, de tal manera, que conjuntamente y al unir esas capacidades y conocimientos pudieran cumplir al logro de determinadas tareas. Las ventajas observadas fueron la capacidad de distribuir el conocimiento, la facilidad de admitir nuevos agentes con conocimientos diferentes y contar con un sistema robusto en caso de ocurrir algún error.

Los agentes también aparecieron en la computación gráfica. La animación por computadora produjo películas con actores virtuales que trabajan con determinado comportamiento programado previamente de manera manual [3]. Además, investigaciones sobre la vida artificial centran su atención en actores autónomos y su comportamiento. Empresas como Microsoft incluyen varios agentes que cumplen con el objetivo de ayudar a los usuarios en sus tareas.

Los agentes de software se han popularizado en los últimos 12 años [8]. Han sido utilizados para innumerables situaciones. Recientemente, las capacidades de los agentes han sido aplicadas en el comercio electrónico, prometiendo una revolución en la forma de conducir las transacciones.

2.3. Comercio electrónico con agentes

El comercio electrónico incluye una variedad de tareas como son: seguridad, mecanismos de pago, catálogo, promociones y compras por mencionar algunas. La tecnología de agentes puede aplicarse algunas de estas tareas donde es deseable que los sistemas que desarrollen estas tareas posean un comportamiento personalizado, continuo y semi-autónomo. Los agentes pueden automatizar la búsqueda de un producto en las mejores condiciones de venta mediante numerosos vendedores simultáneamente. El uso de los agentes ayuda al cliente a disminuir su búsqueda a través de múltiples tiendas en línea

y aumenta la posibilidad de adquirir un mayor número de ofertas. Los agentes realizan comparaciones en línea más eficientes que las realizadas convencionalmente. Algunos otros agentes desarrollados en sistemas de comercio electrónico, sugieren al cliente productos basados en las preferencias de otros clientes con gustos similares.

La ayuda que proporcionan los agentes en las negociaciones comerciales es profunda. Algunas de las investigaciones relacionadas con el uso de agentes en el comercio electrónico, indican que estos agentes mejoraron las ganancias obtenidas en los mercados electrónicos que los implementaron. Además de las ventajas obtenidas para los portales de comercio electrónico, los usuarios que utilizan sistemas de comercio electrónico basados en agentes obtienen un sentimiento de confianza y satisfacción al percibir que obtuvieron mejores resultados en su búsqueda que los obtenidos por ellos mismos.

Con el objeto de mostrar las ventajas que proporcionan los agentes en sistemas de comercio electrónico se plantea el siguiente ejemplo. Actualmente, es posible establecer compras de productos vía Internet. Esta compra se completa cuando el cliente recibe los productos en su hogar. Los clientes encuentran en estas compras grandes ventajas, los productos de su preferencia llegan a su hogar sin que exista la necesidad de ir por ellos. Además de ello, pueden realizar sus compras a la hora de su preferencia ya que no hay restricciones en cuanto a los horarios que este tipo de tiendas virtuales ofrece. Los vendedores reducen notablemente los costos que implica alquilar un local.

Pese a que las ventajas son grandes, aún existen ciertas desventajas. Los clientes se ven en la necesidad de ir comparando precios en las diferentes tiendas virtuales o en su defecto buscar las mejores ofertas. Ésta es una tarea que aunque puede ahorrar dinero también puede ser demasiado tardada. Aparte de estos problemas se pueden presentar otros más. En algunas ocasiones los clientes desconocen las direcciones de las tiendas virtuales donde pueden adquirir los productos. Los clientes normalmente realizan sus compras a la hora en que la red está demasiado saturada, logrando con ello un grado de lentitud en la búsqueda de productos. Aunado a ello, cada tienda virtual dispone de su propia interfaz con el usuario complicando sumamente el acceso.

Ahora supongamos que existe una herramienta que puede establecer una búsqueda que obtiene los mejores resultados tanto en precio como en calidad de productos, con el simple hecho de indicar los productos que se van a buscar. La herramienta además ofrece la opción de repartir las compras entre diferentes tiendas para conseguir los mejores precios. Esta herramienta consiste en un **agente inteligente de compra**. Este agente no tiene horario alguno, puede realizar sus compras a la hora que no exista saturación en la red. Como usuarios de agente, sólo necesitamos conocer su funcionamiento sin

necesidad de preocuparnos por la interfaces. Como podemos observar, los beneficios que podríamos obtener al desarrollar este agente serían enormes. Los clientes obtendrían los mejores precios en un menor tiempo y los vendedores aumentarían sus ventas sin necesidad de contar con un local establecido trabajando en una jornada de tiempo limitada.

2.4. Tipos de agentes

Los sistemas de comercio electrónico proporcionan desde pequeños servicios de compra en línea hasta infraestructuras complejas que incluyen una gran cantidad de servicios. Estos sistemas se caracterizan por desarrollar una o varias de las etapas que comprende el proceso de compra:

1. Identificación de las necesidades del comprador
2. Elección del producto a buscar
3. Elección del lugar donde se establecerá la compra (vendedor)
4. Negociación
5. Compra y envío
6. Uso y evaluación del producto comprado

Los agentes que ayudan a desarrollar estas etapas son nombrados de la siguiente forma:

- a. **Agentes notficadores:** Son aquellos agentes que tienen la finalidad de notificar a los clientes la existencia o aparición en el mercado de productos adecuados a sus preferencias o necesidades. Un ejemplo de este tipo de agentes es el denominado *Jango* (<http://www.jango.com>).
- b. **Agentes de recomendación:** Su objetivo es realizar recomendaciones a los usuarios de los productos que pudieran ser de su interés. Para dar estas recomendaciones, los agentes se basan en el perfil del cliente. Un ejemplo de agente de recomendación es *Firefly*.

- c. **Agentes de compra comparativa:** Los agentes de compra comparativa tienen el fin de localizar a los comerciantes que ofrecen las mejores condiciones de compra de un producto determinado. Ejemplo de agentes de compra comparativa son: *Bargain Finder* y *ShopBot*.
- d. **Agentes de negociación:** Tratan de llevar al mercado electrónico hacia un proceso de negociación que se desarrolla a la hora de establecer una relación comercial. Este tipo de agentes pueden clasificarse en agentes compradores y agentes vendedores que interactúan con la finalidad de llegar a un acuerdo en el que establecen las condiciones de adquisición de un producto determinado. Un agente que implementa negociaciones es *Kasbah* [1].

A continuación, se describe el funcionamiento general de los sistemas de comercio electrónico que emplean algunos de los agentes anteriormente mencionados. Se define características generales y funcionamiento.

Firefly

FireFly de Agents Inc. es un ejemplo de un agente de recomendación. FireFly aplica varias encuestas e índices de popularidad y tomando como base los resultados obtenidos determina las preferencias musicales y de artistas formando una lista de discos compactos en la que residen estas preferencias. Un ejemplo de funcionamiento general de Firefly es el siguiente. Un cliente introduce la referencia de un cantante de rock tras unos minutos el sistema devolverá una lista de los discos no sólo de este cantante de rock sino de otros más que corresponden al mismo género. El cliente recibe los discos de los cantantes que el agente está recomendando basándose en sus preferencias musicales.

Firefly realiza un filtrado automático para recomendar productos a los usuarios. Firefly es un sistema que ha tenido un gran éxito. Cabe mencionar que *Yahoo* (<http://www.yahoo.com>) usa el agente Firefly para personalizar su generador de listas con los diez mejores éxitos de música y películas.

Bargain Finder

Bargain Finder de Andersen Consulting, es un agente cuyo principal objetivo es el de conseguir de entre varias tiendas de música los menores precios de discos compactos y cintas. El sistema es, en esencia, un motor de búsqueda de bases de datos. Los clientes que desean hacer uso de este agente deben conocer el nombre del artista y álbum que desean encontrar. De esta manera, podemos decir que el sistema se inclina por los clientes

que saben exactamente lo que quieren en lugar de compradores ocasionales. Las empresas que desean aparecer como principales proveedores de productos deben suscribirse pagando una cuota por aparecer como referencias para los clientes.

Jango

Jango es un agente que notifica productos en Internet. Hace uso de los servicios del buscador *Excite* (<http://www.excite.com>). El funcionamiento del sistema en esencia es el siguiente: el usuario proporciona el nombre del producto y Jango determina cuáles son las tiendas y los sitios relevantes en los que el cliente puede encontrar ese producto. Jango consulta los sitios y muestra información al cliente sobre datos del producto como su precio y sus especificaciones. Jango se fundamenta en la recuperación de información en paralelo de la Web.

Firefly, Bargain Finder y Jango son hoy en día sistemas comerciales de gran éxito en la compra y venta de productos. En este documento no se hacen referencias técnicas de ellos ya que no existe mucha información debido a su carácter comercial.

Kasbah

Kasbah es un sistema Web en el cual los usuarios pueden crear agentes autónomos para que realicen las tareas de comprar y vender productos [1]. Este sistema ha sido desarrollado con la finalidad de facilitar al usuario las tareas de búsqueda y elección de un producto y, además de ello, evitar que el usuario se involucre en la tediosa tarea de negociar por la obtención del mejor precio posible.

El prototipo Kasbah incluye agentes vendedores, compradores y un mercado virtual. Los agentes compradores y vendedores negocian en el mercado Kasbah usando el mismo lenguaje. Kasbah funciona en términos generales de la siguiente forma. Cuando un usuario desea vender un producto crea una agente vendedor al que le indica la descripción del producto que pondrá en venta. Este agente se introduce en el mercado virtual anunciando su producto. El agente debe contactar con los agentes compradores y negociar con ellos con el objeto de conseguir el mejor precio posible.

El agente vendedor es autónomo y tiene la capacidad de negociar con los agentes compradores sin intervención del usuario. Pese a ello, el usuario tiene finalmente el control del agente ya que al crearlo le da los parámetros en los que establece la negociación. Los parámetros que el usuario da al agente vendedor son: fecha deseada de venta, precio deseado y precio mínimo. El agente vendedor toma sus propias decisiones siguiendo estos parámetros. Estos parámetros pueden ser modificados por el usuario en el momento en

que así lo solicite.

Las heurísticas utilizadas por los agentes vendedores en la negociación son las enumeradas a continuación:

1. Iniciar la oferta del producto tomando el parámetro del precio deseado por el cliente y en caso de existir compradores completar la venta.
2. Si no hay compradores, el precio del producto disminuirá ligeramente con el objeto de despertar el interés de los compradores.
3. Si el período de venta el artículo está por finalizar y éste no se ha vendido, se fijará como precio de venta el precio mínimo aceptable.

Puede darse el caso de que no existan agentes compradores interesados en el precio del producto del agente vendedor, por la razón de que este precio sea demasiado elevado. Si esto ocurre el agente vendedor fracasará al tratar de establecer su venta. El usuario siempre tendrá el control de sus agentes. Cuando el agente vendedor consigue establecer la venta del producto con algún agente comprador, el usuario puede aceptar o rechazar la oferta antes de que la compra finalice. Para lograr esto, el usuario tiene dos controles adicionales que comienzan cuando se crea el agente: obtener la aprobación del usuario antes de terminar el trato y enviar una notificación por correo al usuario cuando se establezca el acuerdo.

Los agentes compradores se construyen de manera similar a los agentes vendedores. Al construir un agente comprador el usuario debe proporcionarle una descripción del producto que pudiera ser de su interés, dando la posibilidad de indicar al agente comprador que realice la compra en un conjunto específico de agentes vendedores.

El agente comprador recibe del usuario que lo crea los siguientes parámetros que lo ayudarán a establecer su compra: fecha de compra, precio deseado, precio máximo aceptable. Una vez que el agente de compra es creado, al igual que el agente vendedor, se libera en el mercado virtual. Cuando los agentes comprador y vendedor llegan a un acuerdo consiguiendo la aprobación de sus usuarios, entonces se realiza la transacción física de productos con la intervención de ellos.

Para probar el prototipo Kasbah se realizaron pruebas en donde los usuarios compraban y vendían cartas. El objetivo de cada usuario era maximizar la calidad de su mano de cartas siguiendo reglas de póker. El intercambio se realizó con dinero virtual. Inicialmente, los usuarios tenían una cantidad de dinero fija estando incentivados para crear agentes compradores y vendedores.

2.5. Otros agentes

Como definimos en la sección anterior, los agentes pueden utilizarse para automatizar muchas de las tareas involucradas en el proceso de compra que se lleva a cabo en los sistemas de comercio electrónico, obteniendo mediante su uso grandes beneficios. Además de las aplicaciones descritas anteriormente, encontramos en Internet agentes que desarrollan otras tareas:

- **Guías de Viaje:** Los usuarios que navegan en Internet en algunas ocasiones se pierden o desorientan debido a la inmensa cantidad de páginas Web existentes. Los agentes actúan como guías de viajes orientando al usuario sobre los vínculos por los que puede seguir. Un ejemplo de este agente es *WebWatcher* (<http://www.webwatcher.com>). Este agente se encarga de aconsejar al usuario sobre los hipervínculos por los que puede seguir después y aprender sobre las reacciones del usuario.
- **Agentes de indexamiento de documentos:** Estos agentes se encargan de almacenar en forma exhaustiva índices de palabras de títulos de documentos y de textos de documentos. Para encontrar estos documentos realizan una búsqueda autónoma en la Web. El usuario pregunta al agente por un documento que contenga ciertas palabras y el agente muestra todos los posibles resultados de acuerdo a los documentos almacenados previamente. Ejemplos de este tipo de agentes son *Lycos* (<http://www.lycos.com>), *WebCrawler* (<http://webcrawler.com>) e *Infoseek* (<http://go.com>).

Actualmente, se desarrollan numerosos sitios Web que utilizan agentes debido a los grandes beneficios que otorga el uso de esta tecnología. Explicamos en esta sección dos aplicaciones más de los agentes, estas aplicaciones son diferentes a las que se establecen en una compra. En este trabajo de investigación abordamos el desarrollo de un agente que establece vínculos comerciales entre clientes y diversas empresas. El agente puede considerarse como *agente de compra comparativa* por el tipo de operaciones que realiza. Desarrolla tareas que comprenden el proceso de compra (ver sección 2.4).

2.6. Servicios Web

Para el desarrollo del agente de búsqueda definido en este trabajo de investigación, se utilizó la tecnología denominada servicios Web. Un *servicio Web* es esencialmente la infraestructura de comunicaciones que permite invocar un método público ofrecido por un servidor [9]. Es un componente de software que puede ser accedido desde Internet para su empleo en otras aplicaciones que se encuentren en ambientes distribuidos. Los servicios Web definen mecanismos estandarizados que permiten describir, localizar y comunicar aplicaciones en línea.

Los servicios Web se encuentran divididos en 3 áreas: protocolos de comunicación, descripción y descubrimiento de servicios.

- **Protocolos de comunicación.** El protocolo simple de acceso de objetos SOAP [10] permite la comunicación entre servicios Web.
- **Descripción de servicios.** El lenguaje de descripción de servicios Web WSDL [11] proporciona un conjunto de definiciones que describen un servicio Web.
- **Descubrimiento de servicios.** El directorio de descripción, descubrimiento e integración universal UDDI [12] es un repositorio de descripción de servicios Web.

En las siguientes secciones se abordan brevemente estos temas.

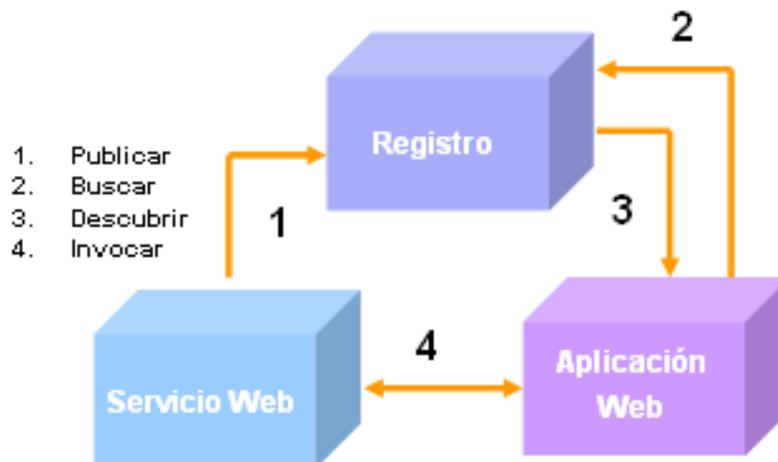


Figura 2.1: Registro e invocación de un servicio Web

Para que un servicio Web pueda invocarse por otras aplicaciones debe crearse y publicarse. En la figura 2.1 se observa de manera general las acciones que afectan el funcionamiento de un servicio Web. En primer lugar el servicio debe crearse, una vez que existe debe publicarse (punto 1 de la figura 2.1). Un servicio Web se publica con la finalidad de que sea conocido y de esta manera pueda invocarse. Invocar a un servicio Web es sinónimo de hacer una petición al servicio. Un servicio Web se publica usando un servicio de directorio distribuido (UDDI) como se describe en la sección 2.11. Cualquier cliente, en este caso una aplicación Web, formula una solicitud de búsqueda hacia un servicio Web que cumpla con sus necesidades (punto 2 de la figura 2.1). El cliente recibe como resultado la dirección donde se encuentra especificado el servicio Web (punto 3 de la figura 2.1). Finalmente el cliente lanza una petición al servicio Web para recibir mas tarde una respuesta (punto 4 de la figura 2.1).

El desarrollo de servicios Web proporciona grandes ventajas. En primer lugar el tiempo de desarrollo se reduce. Al mismo tiempo disminuye la utilización de recursos de cómputo ya que el tratamiento de información que proporciona el servicio se realiza en las máquinas donde se localiza el servicio. Los servicios Web ayudan a establecer interoperabilidad en aplicaciones distribuidas. Además, proporcionan herramientas que ayudan al desarrollo y mantenimiento de aplicaciones.

Los servicios Web logran su funcionamiento adoptando tecnologías como XML, SOAP, WSDL y UDDI.

2.7. HTML

Los documentos en la Web emplean una representación estándar para publicar información conocida como HTML (*HyperText Markup Language*). HTML es un lenguaje que permite describir hipertexto. Hipertexto consiste de un grupo de documentos que pueden contener texto, imágenes, video y audio, así como enlaces a otros documentos [13]. HTML no incluye instrucciones detalladas de formato, simplemente proporciona guías muy generales de presentación. El navegador del cliente se encargará de interpretar estas guías, es por esta razón que las páginas Web pueden verse de manera diferente en ciertos navegadores.

Un documento HTML inicia con una etiqueta `<html>` y finaliza con `</html>`. Dentro de estas dos etiquetas encontramos el documento estructurado en dos partes principales: una cabecera y un cuerpo. La cabecera contiene los detalles del documento y el cuerpo la información más importante. La cabecera utiliza las etiquetas `<head>` y

`</head>`, dentro de las cuales se encuentra su contenido. El título del documento HTML forma parte de la cabecera y se encuentra entre las etiquetas `<title>` y `</title>`.

El cuerpo del documento HTML se localiza después de la cabecera entre las etiquetas `<body>` y `</body>`. El cuerpo es en donde se encuentra el texto con la presentación y los efectos que decida poner el autor. El cuerpo del documento contiene la información que el lector podrá ver en su navegador. El cuerpo de un documento HTML puede contener diversas etiquetas que pueden incluir imágenes, tablas, video, colores, etc. El autor decide cuáles, cómo y en qué lugar colocarlas con el objeto de dar una mejor presentación a este documento.

HTML se ha convertido en uno de los instrumentos principales para marcar documentos suministrados a través de la Web. Entre algunas de las ventajas que presenta este lenguaje se pueden mencionar las siguientes: posee una sintaxis simple con un conjunto fijo de etiquetas, ofrece la posibilidad de enlazar varios documentos entre sí y permite crear de manera sencilla documentos multimedia incorporando imágenes y sonido. Pese a las ventajas que ofrece el lenguaje, una de sus principales desventajas radica en que no contiene etiquetas para representar datos lógicos, sino que únicamente utiliza un conjunto fijo de etiquetas para presentar la información que compone el documento.

2.8. XML

El lenguaje extensible de etiquetado *XML* (*Extensible Markup Language*) es un conjunto de reglas para definir etiquetas semánticas que organizan un documento en diferentes partes [14]. XML es una tecnología que permite a los diseñadores de páginas Web crear sus propias etiquetas personalizadas, obteniendo funcionalidad que no proporciona el lenguaje de marcado que actualmente utilizan la mayoría de las aplicaciones Web (HTML). HTML está diseñado para presentar información, siendo inadecuado para procesar para los programas informáticos. A diferencia de HTML, XML describe el contenido de lo que etiqueta.

Tanto XML como HTML se derivan del lenguaje SGML (*Standard Generalized Markup*), *Lenguaje Estandarizado y Generalizado de Marcado* definido en 1986 como un estándar internacional para el marcado de documentos. XML remonta sus inicios hacia el año de 1997. W3C [15] lanzó en Febrero de 1998 la primera versión, la Recomendación 1.0. W3C es el sitio oficial del consorcio de la World Wide Web (W3C), un consorcio internacional de compañías que tiene la finalidad de desarrollar estándares abiertos con el fin de que la Web evolucione en una sola dirección.

XML es un proyecto del W3C. Cabe mencionar que W3C define estándares relativos a la Web, los cuales no se consideran como un conjunto de normas oficiales. Debido a esto, el consorcio emite su opinión en forma de recomendaciones y no como estándares internacionales. Sin embargo, en la práctica, las recomendaciones de este organismo tienen la misma validez y autoridad que el resto de estándares emitidos por organismos normativos.

XML es un lenguaje de marcado extensible, lo que significa que en XML puede definirse un conjunto personalizado de etiquetas mediante el uso de una DTD (*Document Type Definition*). Una definición de tipo de documento, consiste en la declaración del tipo de elementos, atributos y entidades específicas contenidas en el documento XML. Es decir, una DTD es una colección de etiquetas y reglas que definen la gramática (estructura y contenido) de un documento. Para comprender mejor el concepto de DTD se plantea el siguiente ejemplo.

En un documento XML se necesita incorporar una etiqueta que contenga el nombre de una ciudad. La etiqueta se define como `<Ciudad>`. Esta nueva etiqueta permitirá que otras aplicaciones puedan extraer de ella la información referente al nombre de la ciudad. Para lograrlo, la etiqueta deberá definirse en la gramática del documento, lo que significa que se creará un nuevo documento (DTD) en donde se define el tipo de elemento correspondiente a esta nueva etiqueta. El DTD permite definir el tipo de datos nuevos incorporados el documento XML. La DTD debe publicarse y ponerse a disposición de cualquier persona que desee utilizar el documento. La DTD publicada, actuará como la especificación del documento. En la figura 2.2 se observa un DTD que contiene el elemento definido en el ejemplo anterior.

```
<! ELEMENT Domicilio (Ciudad, Estado, Calle, Número, Manzana, Lote)
<! ELEMENT Ciudad (#PCDATA)>
<! ELEMENT Estado (#PCDATA)>
<! ELEMENT Calle (#PCDATA)>
<! ELEMENT Número (#PCDATA)>
<! ELEMENT Manzana (#PCDATA)>
<! ELEMENT Lote (#PCDATA)>
```

Figura 2.2: Ejemplo de una DTD

En la recomendación 1.0 de XML se especifican algunas de las características que debe contener un documento XML entre algunas de las cuales están: la validez y la buena formación. *Buena formación* en un documento XML se refiere a la conformidad o no con las restricciones de buena formación (WFC) definidas en la Recomendación

1.0 [14]. Algunas restricciones que determinan la buena formación de un documento son: la correspondencia entre etiquetas, nombres de atributos únicos en el documento y los atributos no deben contener el carácter "<". Los documentos que no tengan una buena formación serán rechazados por el procesador de XML.

Validez en un documento XML se refiere a su conformidad o no con las restricciones de validez (VC) especificadas en la Recomendación 1.0. Para conocer si un documento XML es válido o no lo es, se debe incluir la declaración `<!DOCTYPE>` donde se especifica la DTD a la que debe adecuarse el documento para que resulte válido.

Si se tiene las siguientes declaraciones en un documento XML:

```
< ! DOCTYPE Domicilio SYSTEM
"http://www.domicilio.com/Domicilio.dtd">
```

se supone que el URL hace referencia a la ubicación de la DTD. Si en el documento no contiene la declaración `<!DOCTYPE>`, el procesador de XML no comprobará la validez del documento. El procesador XML es el encargado de leer la DTD del documento y comprobar su validez de acuerdo con las restricciones de validez. Para que un documento sea declarado como válido, las CV exigen que todas las etiquetas se hallen definidas dentro de la DTD. La validez se centra en la estructura lógica de los datos, mientras que la buena formación se centra en la física.

Debido a que XML es altamente flexible y potente, se puede utilizar para otras aplicaciones. Entre algunas de las aplicaciones que puede ofrecer XML están:

- XML como descriptor de metacontenidos. En 1997 XML se consideraba un lenguaje para **metacontenidos**. Entendiéndose por este concepto como la información relativa contenido de un documento. Esta información puede corresponder al título, autor, tamaño del archivo, fecha de creación, historial de cambios, etc. XML está considerado como el mejor vínculo para definir la sintaxis de metacontenidos dada su flexibilidad, legibilidad y capacidad de ampliación.
- XML para publicar e intercambiar información de bases de datos. XML conserva información original de los datos como nombres de columnas y tipos de datos, el cliente puede usar esta información para diversos fines, además observarlos en pantalla. Por ejemplo, podría utilizar los datos y descargarlos en una hoja de cálculo.

- XML como formato de mensajes para la comunicación entre programas de aplicación. El área de aplicación más importante de XML es la de **mensajería**. Por mensajería se entiende el intercambio de mensajes entre organizaciones o entre sistemas de aplicación dentro de una misma organización [14].

2.9. SOAP

SOAP (*Simple Object Access Protocol*) es un protocolo de mensajería XML que forma la base de los Servicios Web. SOAP define un mecanismo que ofrece la posibilidad a una aplicación de enviar mensajes XML a otra. Permite el intercambio de información en un ambiente distribuido [16]. EL diseño de SOAP trata de resolver los problemas encontrados en el protocolo HTTP en el que su utilización está limitada al uso de comandos simples (GET, POST y PUT) lo que impide el intercambio de datos de forma libre entre aplicaciones [17].

SOAP es un protocolo basado en XML para la mensajería y la llamada a procedimientos remotos (RPCs). Es completamente independiente del protocolo de transporte subyacente, por eso los mensajes SOAP se pueden intercambiar sobre HTTP, JMS o protocolos de transporte de e-mail. Actualmente el protocolo HTTP es el más utilizado para los mensajes HTTP. SOAP es un protocolo de alto nivel que sólo define la estructura del mensaje y unas pocas reglas para su procesamiento.

Un mensaje SOAP es una transmisión de va desde un emisor SOAP hacia un receptor SOAP, cabe mencionar que cualquier aplicación puede adoptar el papel de emisor o de receptor. Los mensajes SOAP pueden combinarse de diversas formas, de manera que permiten combinar diversos comportamientos de comunicación como lo son solicitud/respuesta, respuesta solicitada, mensajería asíncrona de una-vía o incluso notificación. Un mensaje SOAP contiene una estructura relativamente sencilla compuesta por un elemento XML con dos secciones, una de ellas corresponde al encabezado y otra al cuerpo. La figura 2.3 muestra la estructura de un sobre SOAP.

SOAP puede usarse de dos formas: como protocolo simple de mensajería ó como protocolo que permite definir llamadas a procedimientos remotos. SOAP como un protocolo simple de mensajería, crea y envía mensajes SOAP hacia un servicio para su procesamiento. Para utilizar SOAP en llamadas a procedimientos remotos, se debe definir un protocolo que incluya la siguiente información:

1. Primeramente, se debe definir la forma en que los valores (la información) será transportada entre SOAP y las aplicaciones, y

```
<SOAP: Envelope xmlns: SOAP=
    "http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP: Header>
    <!-- El contenido del encabezado se sitúa aquí-->
  </SOAP: Header>
  <SOAP: Body>
    <!-- el contenido del cuerpo va aquí -->
  </SOAP: Body>
</SOAP: Envelope>
```

Figura 2.3: Estructura de un mensaje SOAP

2. Los objetos, operaciones y parámetros hacia donde las RPCs deben llegar.

La diferencia entre ambas formas radica en que la primera de ellas solo envía un mensaje hacia un servicio y espera obtener de él una respuesta. La segunda llama a un método remoto el cual puede solicitar parámetros para realizar su procesamiento, es por esta razón que el mensaje contendrá el método hacia donde se envía el mensaje y los parámetros que este solicita si es el caso.

2.10. WSDL

El lenguaje de descripción de servicios Web *WSDL* (*Web Services Description Language*) es un formato XML desarrollado por IBM y Microsoft para describir servicios Web como una colección de puntos de comunicación que pueden intercambiar ciertos mensajes [16]. WSDL especifica la descripción funcional y técnica del servicio Web. En la especificación del servicio Web podemos encontrar lo que hace el servicio Web. De igual forma, se pueden identificar los métodos que están disponibles en el servicio y los tipos de parámetros. Un programador debe conocer como está especificado un servicio Web para poder crear un cliente que sea capaz de invocarlo.

WSDL describe los servicios Web especificando los mensajes que se intercambian entre el proveedor de servicio y el cliente. Un documento WSDL está compuesto de 3 partes básicamente.

La primer parte contiene los elementos *types*, *message*, y *portType*. En esta primer parte se define el mensaje y los tipos de datos que se intercambian entre el cliente y el servidor. Un *message* puede contener una o más partes, en ellas se especifica un parámetro con tipo. Los mensajes pueden ser mensajes de entrada o de salida para cada

método de la clase Java. Los mensajes se agrupan en *operations* en una entidad llamada *portType*. Un *portType* consiste en un conjunto de operaciones que son colectivamente soportadas por un servicio Web.

En la segunda parte se definen los elementos de vinculación (*binding*). En ellos se especifican los detalles de la implementación técnica del servicio Web. El elemento *binding* describe el protocolo de comunicación que se utilizará (como SOAP sobre HTTP) y la forma de cómo se llevarán a cabo las interacciones en el protocolo. En él se define el protocolo de comunicación y el formato de especificación de un *portType*. La última parte se define en el elemento *service*. Este elemento indica donde finaliza la comunicación (el URL). El URL indica la dirección de red a donde se encontrará especificado el servicio Web.

La importancia de WSDL es alta ya que junto con UDDI especifican el URL al que se invocará el servicio y el URL de la especificación del servicio. El URL de invocación del servicio contiene la información necesaria que permite invocar el servicio Web. El URL de la especificación contiene la información necesaria del servicio referente a su descripción y especificación.

2.11. UDDI

UDDI (*Universal Description Discovery and Integration*) es un repositorio de descripción de servicios Web. UDDI es una base de datos universal de servicios Web. En UDDI los desarrolladores pueden buscar cualquier sitio para descubrir servicios Web. UDDI ofrece a los usuarios una forma sistemática de encontrar servicios mediante un registro centralizado de servicios que puede ser equivalente a un directorio de servicios Web en línea [16].

Para registrar un servicio Web mediante UDDI se utiliza el servicio Web de publicación de UDDI, que brinda operaciones como el almacenamiento y eliminación de negocios. La información de las compañías se almacena en un documento XML. Para que cualquier cliente pueda descubrir el servicio Web de una compañía se utiliza el servicio Web de pregunta de UDDI. Este servicio devuelve al cliente que así lo solicitó una lista de las compañías que están registradas en UDDI y que satisfacen una condición determinada.

UDDI proporciona 2 especificaciones básicas que definen la estructura del registro de servicio y operación:

- La información correspondiente a cada servicio y la forma de codificarlo

- La descripción de cómo esta información puede ser utilizada y actualizada

En la figura 2.4 se muestra el funcionamiento general de UDDI aplicado a un ejemplo. Primeramente, una compañía de software registra sus servicios Web en UDDI. Una aplicación comercial solicita a UDDI la especificación de un servicio que cumpla con sus necesidades. UDDI devuelve a la aplicación comercial la dirección URL donde se encuentra la especificación del servicio solicitado. La compañía finalmente invoca al servicio Web de la compañía.

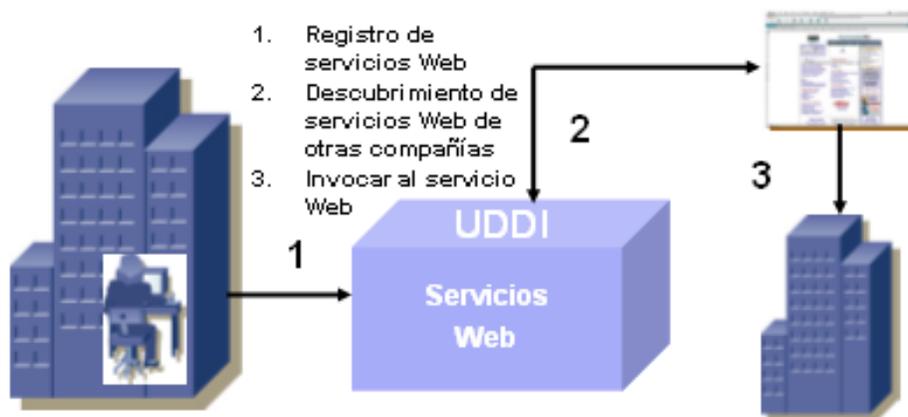


Figura 2.4: Funcionamiento general de UDDI

2.12. Resumen

En este capítulo se explicaron conceptos básicos de agentes de software. Definimos diferentes tipos de agentes que desarrollan alguna de las actividades involucradas en el proceso de compra de productos. Se explicaron también las ventajas que se consiguen al utilizar agentes en sistemas de comercio electrónico. Ejemplificamos algunos sistemas de comercio electrónico que fundamentan su funcionamiento en el uso de agentes. Como se mencionó a lo largo del capítulo, el sistema que se describe en esta tesis fue desarrollado utilizando un agente que realiza la búsqueda de productos en diferentes tiendas. Para el desarrollo del agente se utilizó la tecnología de servicios Web. Asimismo se explicó el funcionamiento básico de un servicio Web. Describimos mediante un diagrama como se lleva a cabo la creación, publicación e invocación de un servicio Web. Finalmente, se detallaron las tecnologías involucradas en el funcionamiento de un servicio Web XML, SOAP, WSDL y UDDI.

Capítulo 3

Caso de estudio: Búsqueda de productos en STD

En este capítulo se describe la funcionalidad del sistema desarrollado en este trabajo de investigación mediante un caso de estudio. Se expone cada una de las acciones que debe seguir un usuario del sistema para lograr su objetivo, que es la compra de productos. Para lograrlo, se plantea la compra de una lista de productos en un conjunto de tiendas. El sistema tendrá previamente los datos necesarios para responder a las solicitudes de los clientes. Dicho de otra forma, la infraestructura del sistema está totalmente abierta para que cualquier usuario pueda utilizarlo.

3.1. Consideraciones iniciales

Antes de abordar el caso de estudio, determinaremos las condiciones iniciales en las que se encuentra el sistema antes de comenzar a utilizarlo así como el escenario en el que se desarrolla el caso de estudio. Para su buen funcionamiento, el sistema debe tener registrada por lo menos una tienda, la cual proporcionará la información de los productos que finalmente comprará el cliente. Para fines de probar la utilidad del sistema se crearon y registraron 3 tiendas, cada una de las cuales posee una base de datos en la que almacena sus productos.

El sistema contará con la información aportada por la tiendas, mostrando finalmente al cliente los datos de productos contenidos en ellas.

Escenario

El escenario donde se desarrolla el caso de estudio es el siguiente. Un usuario se ve en la necesidad de adquirir una determinada cantidad y tipo de productos. El usuario sabe que los productos que necesita se localizan en diferentes tiendas. Pese a ello, no sabe que productos se venden con exactitud en cada una sin mencionar que no sabe dónde puede encontrar los mejores precios. Por otra parte, estas tiendas se encuentran separadas geográficamente lo que provocaría que usuario gastara gran cantidad de tiempo en buscar en cada una de ellas hasta encontrar los mejores precios, cosa que no puede lograr ya que dispone de tiempo limitado. El usuario intenta realizar entonces su búsqueda vía Internet observando una situación similar. Existe una gran variedad de sitios de comercio electrónico que ofrecen diversos productos. El usuario necesita ahora buscar en cada uno de ellos comparando precios hasta decidir cuales son la mejores opciones, situación que puede requerir de igual forma bastante tiempo.

El sistema *STD* desarrollado en esta tesis, ofrece una solución a los problemas anteriores. *STD* es un portal Web que ofrece a los usuarios la búsqueda de los productos que necesitan, con la ventaja de que no tienen que trasladarse físicamente a diferentes tiendas para encontrar los productos ni visitar diferentes portales de comercio electrónico buscando los mejores precios. Además de las ventajas que representa el evitar el traslado físico a sitios apartados, el usuario solamente requiere conocer un solo portal en lugar de tener que conocer el que corresponde a cada tienda. Para que *STD* pueda realizar la búsqueda, solo necesita conocer los nombres de los productos, las cantidades y las tiendas en donde se realizará la búsqueda. Al finalizar la búsqueda, *STD* muestra el cliente una lista de los productos que le solicitó indicando en que tienda se encontraron y a que precio. El cliente puede elaborar finalmente una lista de los productos que comprará. *STD* le mostrará el monto total del pago antes de que el usuario decida terminar su compra.

3.2. Búsqueda de productos en STD

Para demostrar la utilidad del prototipo desarrollado en este trabajo de tesis, planteamos el siguiente caso de estudio que describe las actividades que realiza/sigue un usuario del sistema.

ACTIVIDADES

1. El usuario ingresa al portal.
2. Debido a que es un usuario nuevo se registra.
3. Este usuario revisa los diferentes productos que ofrece STD y en base a ello elabora su pedido que contiene los productos de su preferencia.
4. STD pide al usuario escoger la tiendas en las que buscará los productos.
5. El cliente solicita la búsqueda del pedido a STD.
6. STD contacta a las diferentes tiendas de preferencia del usuario, pidiendo a cada una la búsqueda de los productos del pedido elaborado por el usuario.
7. Terminada la búsqueda, las tiendas remitirán el resultado a STD.
8. STD muestra los resultados de la búsqueda al cliente.

En la figura 3.1 se puede observar de manera más clara los pasos que se desarrollan en este caso de estudio.

3.2.1. Bienvenida al sistema

El usuario comienza a utilizar el sistema en el momento en el que mediante el uso de su navegador entra a la página inicial (ver figura 3.2.1). Esta página, incluye un menú en la parte superior que contiene varios enlaces, todos ellos representados mediante botones. El nombre de cada enlace hace referencia a la actividad que desarrolla a su ingreso, los enlaces podrán observarse en cualquiera de las páginas del sistema.

Además de los enlaces, la página de inicio contiene una sección con la cual el usuario puede ingresar al sistema. Los usuarios que estén registrados en el sistema podrán utilizarlo proporcionando el nombre y la contraseña dados al momento de su registro. Para llevar a cabo su registro el usuario ingresa al enlace que lleva por nombre *Regístrate*, lo lleva directamente a la página de registro.



Figura 3.1: Caso de estudio: Búsqueda de productos en STD

3.2.2. Registro de usuarios

Los usuarios nuevos deben registrarse para poder utilizar el sistema. Para efectuar el registro se completa un formulario en el que se aportan los siguientes datos: nombre, contraseña, dirección, ciudad, código postal, país, teléfono, e-mail y fax.

Los datos más importantes que no se deben omitir son el nombre y la contraseña, esto debido a que esta será la información que se debe recordar para poder ingresar al sistema y hacer uso del mismo en ocasiones posteriores.

Una vez que el usuario escribe sus datos, el sistema registra su información. El cliente puede ahora ingresar al sistema utilizando el nombre y la contraseña que usó para su registro. En la figura 3.2.2 se pueden observar los datos que el cliente registró en el sistema.

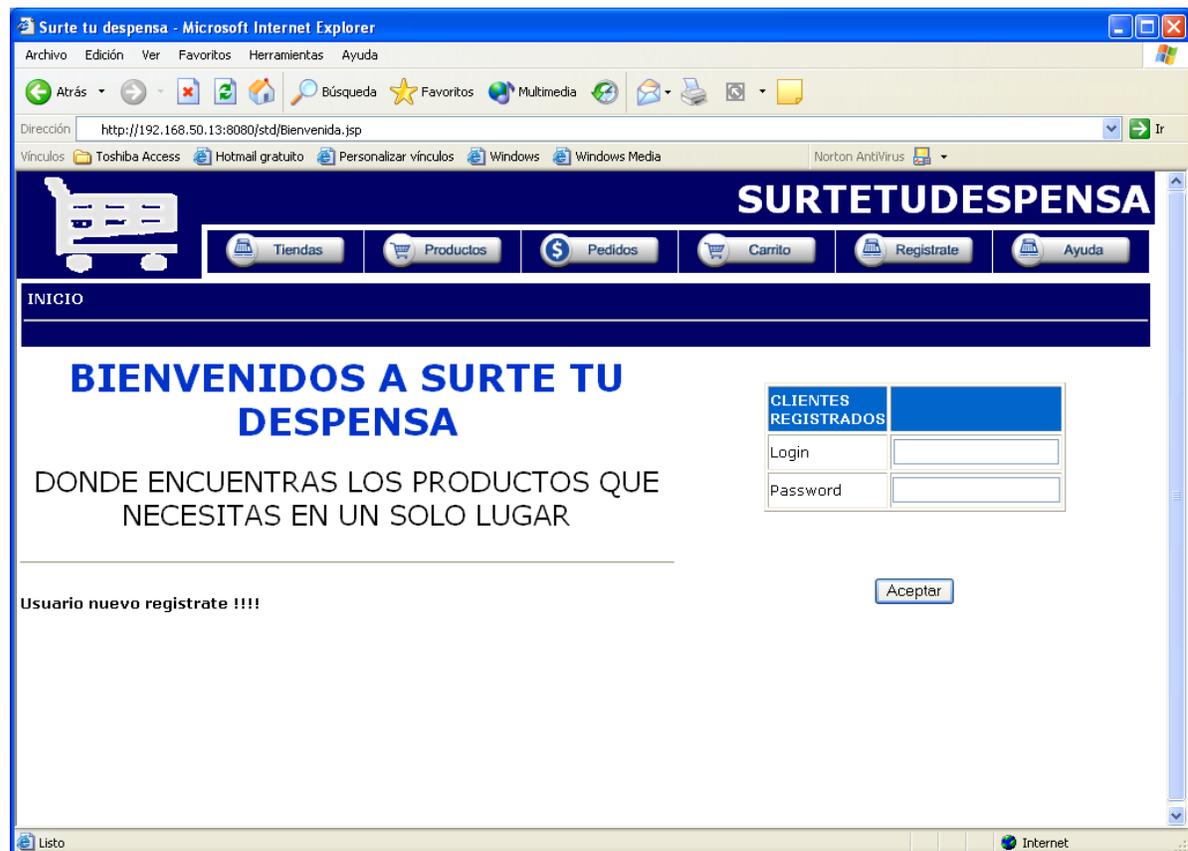


Figura 3.2: Pantalla inicial del sistema

3.2.3. Navegando por los productos

Después de finalizar el registro, el cliente puede entrar al portal en el momento en que así lo solicita. En el menú del Portal existe un botón de enlace que lleva por nombre *Productos*. Este enlace lo lleva a la página de Productos. Para que el cliente pueda solicitar un pedido, primeramente debe escoger los productos que integrarán el pedido. En la página de productos se percibe un listado de la totalidad de productos que ofrece STD. Al entrar a esta página se observa inicialmente una lista de las categorías de los productos. Cada categoría está representada mediante un enlace que muestra las subcategorías de la categoría correspondiente. La mayor parte de las categorías cuenta con una o más subcategorías.

Los productos que integran cada subcategoría se agrupan dependiendo de su tipo al igual que en las categorías. Al entrar al enlace de la subcategoría, se aprecian finalmente los productos que contiene. Las categorías que no contienen subcategorías exponen los productos contenidos en ellas en el momento de su selección.

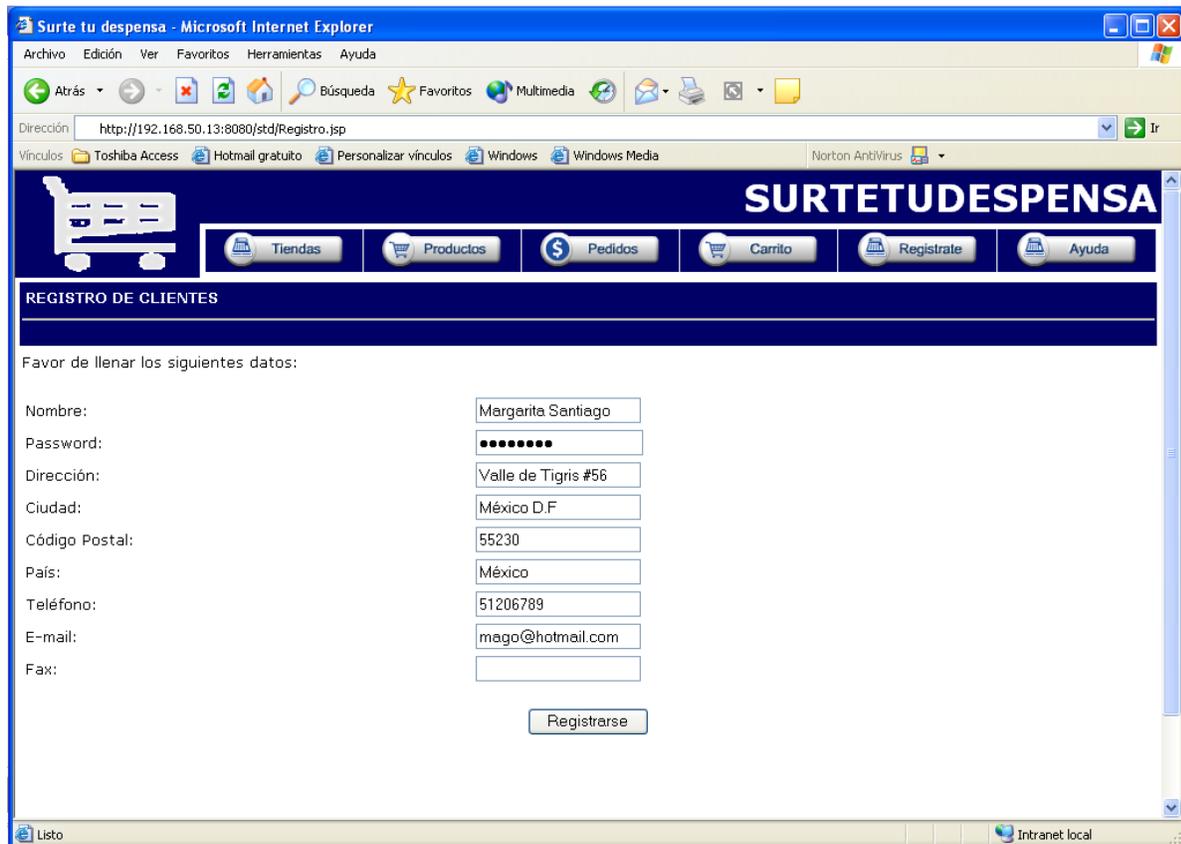


Figura 3.3: Registro del cliente

El cliente puede navegar por los diferentes enlaces de categorías y subcategorías hasta encontrar el producto que necesita. En el momento en que el cliente encuentra el producto puede agregarlo a su pedido al seleccionar el recuadro *Agregar a pedido* seguido de presionar el botón de agregar. Para que el cliente pueda visualizar de manera clara la categoría y subcategoría donde se encuentra situado, en la parte superior de los enlaces de subcategorías y artículos encontrados, respectivamente, se indican los enlaces que ha seleccionado. En la figura 3.2.3 se puede distinguir claramente la forma en que el cliente puede navegar entre los diferentes productos hasta encontrar los de su preferencia y la manera en que puede agregarlos a su pedido. Además, en la parte superior de artículos encontrados están definidas las categorías y subcategorías por las que ha navegado el cliente.

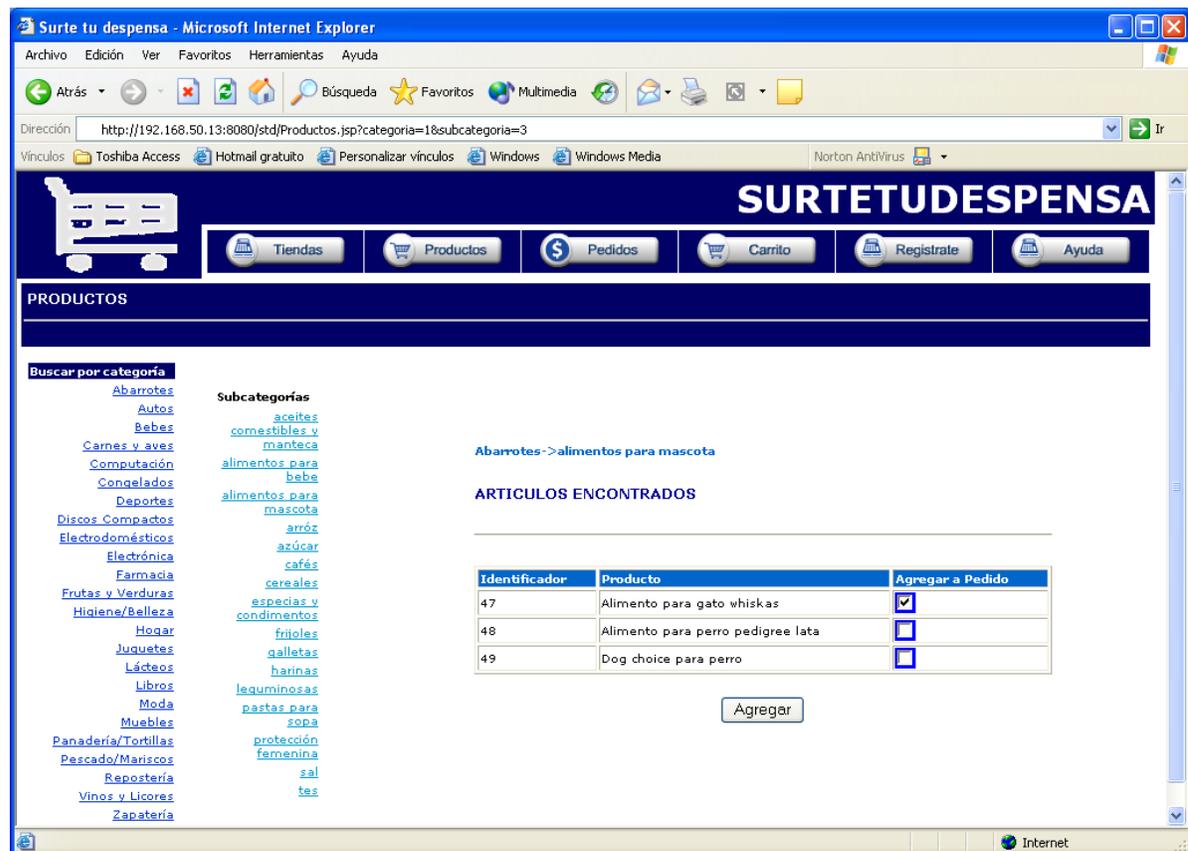


Figura 3.4: Listas de productos

3.2.4. Pedido

En cuanto finaliza con la elaboración de su pedido, el cliente puede ver el contenido de su pedido al presionar el botón de enlace denominado *Pedido*, el cual lo llevará a la página de pedidos. Esta página muestra al cliente una lista de los productos que integran su pedido si es que ha seleccionado alguno. En la figura 3.2.4 se observa la lista de productos del pedido que el cliente escogió en la página de productos.

Como se mencionó en la sección 3.2.3, el cliente puede elegir los productos que contendrá su pedido desde la página de productos. Si el cliente sabe cual es el nombre y/o el código del producto podrá agregar también uno más productos desde la página de pedidos.

En la figura 3.2.4, se observa como el cliente agrega un producto más a su pedido escribiendo en los cuadros correspondientes la información del producto además de la cantidad de unidades que desea buscar. Si no conoce estos datos puede obtenerlos en la página de productos.

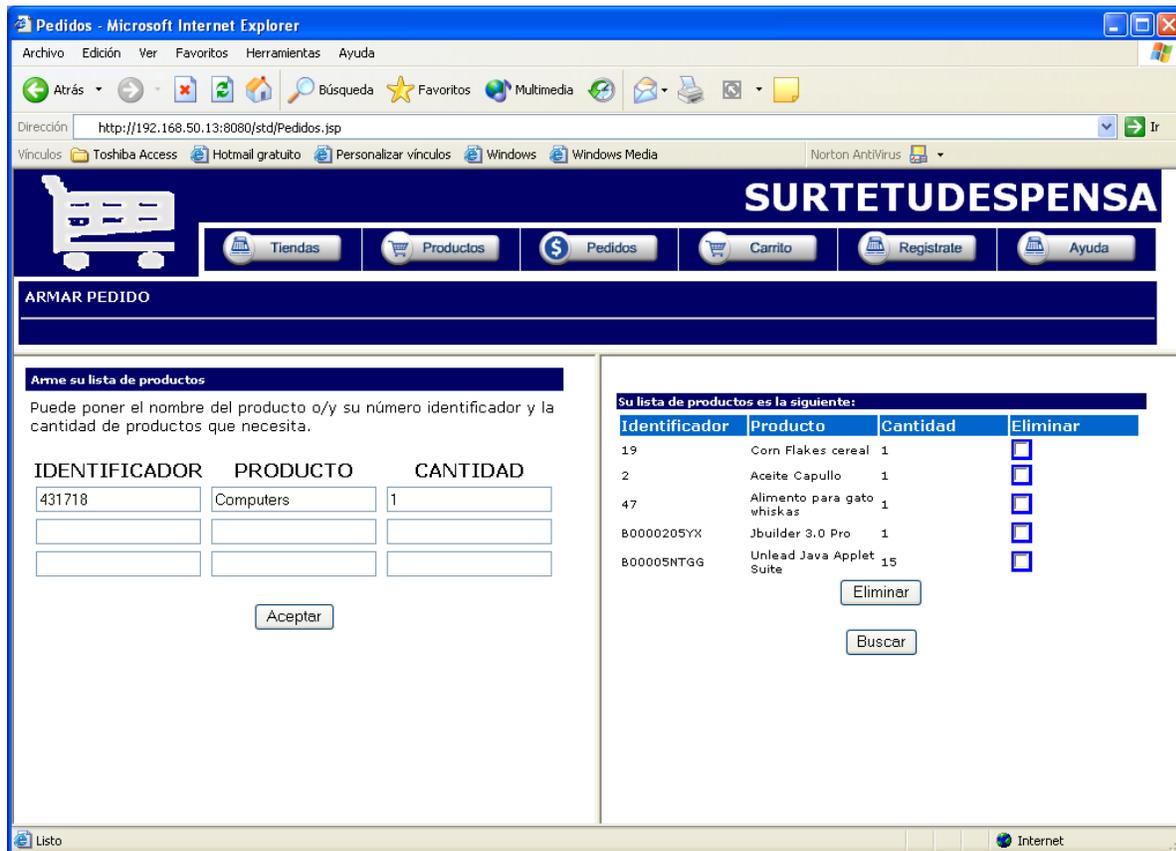


Figura 3.5: Elaboración y visualización del pedido

Para finalizar la creación del pedido, se necesita además, una lista de las tiendas donde se buscarán los productos del pedido. Una vez que el cliente termina de escoger los productos del pedido, presiona el botón de *Buscar* que se encuentra situado en la parte inferior derecha de la página de pedidos (ver figura 3.2.4) que lo lleva a la página de seleccionar tiendas.

La página de seleccionar tiendas suministra al cliente un listado de las tiendas que se encuentran registradas al momento de su petición. En estas tiendas es en donde el sistema busca la existencia y los datos de el(los) producto(s) en la solicitud. Si el cliente no tiene mucho conocimiento sobre estas tiendas, puede obtener más información de ellas al presionar el botón de enlace con el nombre de *Tiendas*.

La página de tiendas contiene la información de cada tienda registrada en el sistema figura 3.2.4. En la primera columna se sitúa el nombre de la tienda, mismo que aparece en la página de selección.

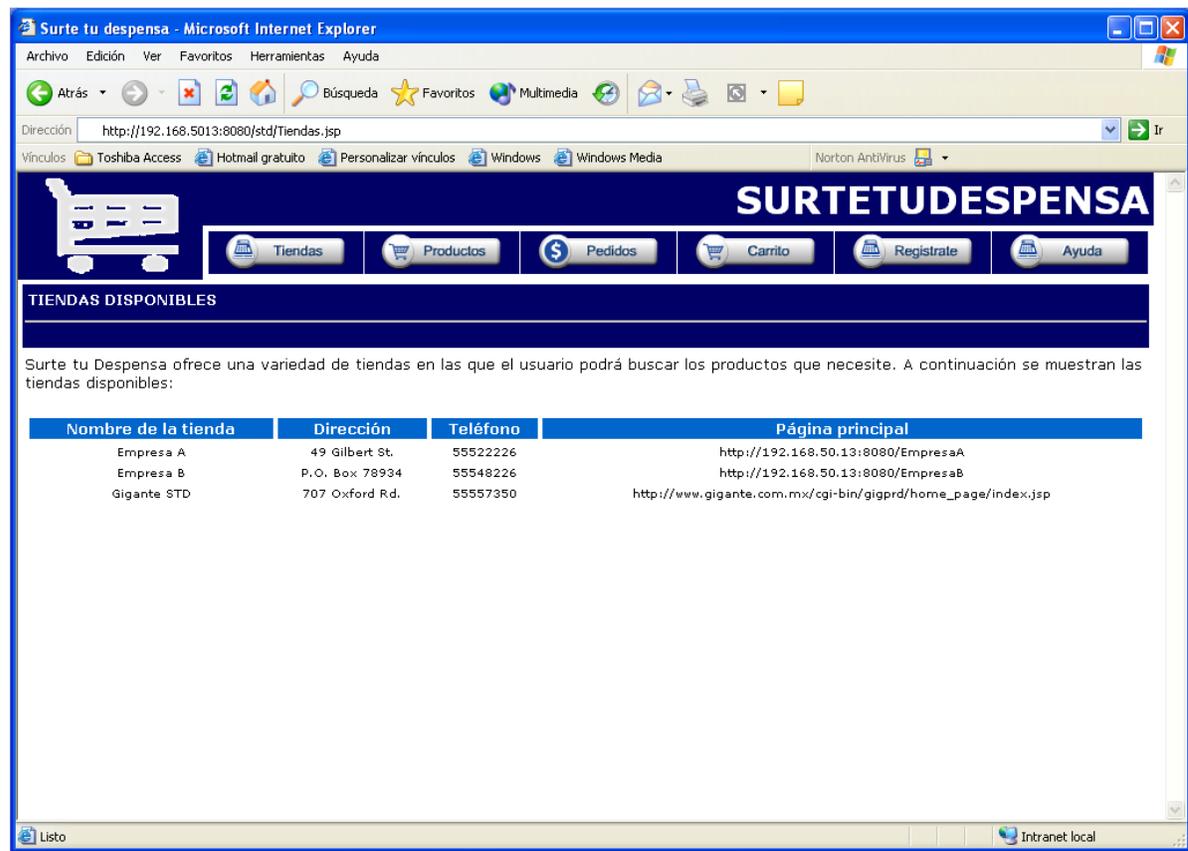


Figura 3.6: Tiendas registradas

La segunda columna contiene la dirección física de la tienda, la tercera el teléfono. La cuarta y última columna posee la dirección Web de la tienda si es que esta cuenta con un Portal.

Las tiendas que se encuentren en la página de tiendas, serán las mismas que aparezcan en la página de selección de tiendas. Como se puede apreciar en la figura 3.2.4 todas las tiendas cuentan con un número situado en la parte derecha de su nombre. El número hace referencia a su prioridad de búsqueda. Dicho de otra forma, el número señala el orden en que se efectuará la búsqueda del pedido en ellas. El número uno es el de mayor prioridad, lo que indica que la tienda que tenga ese número será la primera donde se buscará el pedido. Si el cliente desea omitir alguna de las tiendas en la búsqueda, deberá dejar el espacio correspondiente al número en blanco o con el valor de cero. Los números de búsqueda por omisión serán en orden ascendente, la primera tienda de la lista tendrá el valor de uno.

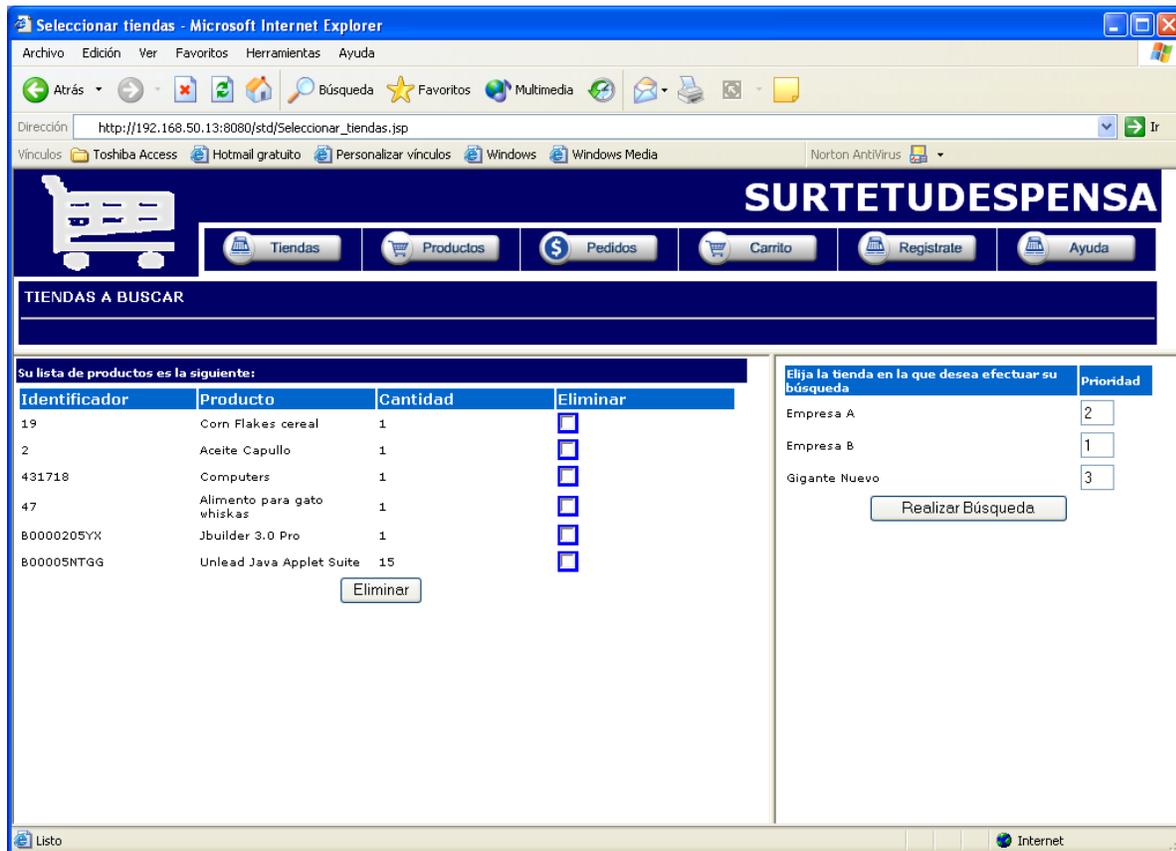


Figura 3.7: Selección de tiendas

En la figura 3.2.4 se indica que la búsqueda se llevará a cabo en el siguiente orden: la primera tienda en la que se realizará la búsqueda será Empresa B, seguida de Empresa A, finalmente se buscará en Gigante STD.

3.2.5. Búsqueda

La etapa de búsqueda de productos comienza desde el momento en que el cliente presiona el botón de *Realizar Búsqueda* situado en la parte inferior derecha de la página de seleccionar tiendas. STD recibe todos los datos del cliente y se prepara para realizar la búsqueda. El sistema manda la solicitud de búsqueda a las tiendas elegidas por el cliente en el orden de prioridad correspondiente. Cada tienda se encargará de procesar las solicitudes de STD y devolver los resultados generados.

En la figura 3.8 se detallan los resultados recibidos por parte de las tiendas a las peticiones de STD. Los resultados de la figura 3.8 se definen siguiendo un orden. Al inicio, STD envía a la tienda correspondiente (de acuerdo a su prioridad) la solicitud

de búsqueda del pedido. Esta solicitud se establece cuando STD invoca al servicio Web denominado *getProductDetails* de la tienda, el cual será discutido en la sección 5.2.2.5. La dirección donde se invoca el servicio Web se define en los resultados de la figura 3.8 como *url*. El servicio Web, recibe como parámetro de entrada un código de producto definido en STD como el número identificador de cada producto, mismo que se define en la figura 3.8 como *Id de Producto*. En respuesta genera los datos de tiempo de entrega, precio y cantidad del producto en solicitud.

Como se puede observar en la figura 3.8, STD envía el pedido solicitado por el cliente a cada una de las tiendas de su elección, que para este caso de estudio son tres. El cliente solicitó a STD que la primera tienda de búsqueda fuera Empresa B. En la figura 3.8 se observa que la primera tienda a la que STD envía la solicitud de búsqueda es la tienda especificada con el número 2, esto se debe a que Empresa B está definida en el sistema con el número de identificador 2.

La tienda 2 buscará cada uno de los productos del pedido, en la figura 3.8 observamos que cada uno de los productos del pedido aparece indicado con su número identificador. Si el producto solicitado se encuentra en la tienda, los resultados de esa tienda serán enviados a STD antes de buscar el siguiente producto. Debido a ello, algunos de los productos de los resultados tendrán al finalizar el *url* los datos de respuesta generados por las tiendas, lo que indica que el producto se localizó en la tienda con los datos especificados en el resultado. Por lo antes dicho podemos inferir lo siguiente, en la solicitud elaborada a la Empresa B el producto con número de identificador 47 no se encontró en la tienda, por lo que sus datos no aparecen al finalizar el *url*. El producto definido con el número de identificador 431718, se encuentra en la Empresa B y contiene los siguientes datos: tiempo de entrega 2 semanas, precio \$60 y cantidad 180 unidades.

El orden en que aparecen los resultados es el mismo orden en que se fueron enviando las peticiones a las tiendas, dependiendo de su prioridad. Los resultados se presentan de igual forma en que se fueron encontrando. Después de finalizada la búsqueda en Empresa B, STD solicitará la búsqueda del pedido a Empresa A, los últimos resultados obtenidos son los de la empresa Gigante STD, ello debido a que fue la tienda que obtuvo la menor prioridad.

3.2.6. Resultados

Una vez que STD obtiene todos los resultados de la búsqueda los proporciona al cliente. La figura 3.2.6 expone los resultados de la búsqueda solicitada por el cliente. Los

```

*****
Comienza la b'squeda en la tienda 2

Se hizo el hilo. Thread[Thread-33,5,main] a la tienda 2
El * Id de Producto * es : 47
El url es: http://192.168.50.13:6080/EmpresaB_1/services/getProductDetails

El * Id de Producto * es : 431718
El url es: http://192.168.50.13:6080/EmpresaB_1/services/getProductDetails

El tiempo de entrega del producto 431718 es : 2 semanas en la tienda : 2
El precio encontrado del producto 431718 es : 60 en la tienda : 2
La cantidad encontrada del producto 431718 es : 180 en la tienda : 2
El * Id de Producto * es : 2
El url es: http://192.168.50.13:6080/EmpresaB_1/services/getProductDetails

El * Id de Producto * es : 19
El url es: http://192.168.50.13:6080/EmpresaB_1/services/getProductDetails

El * Id de Producto * es : B0000205YX
El url es: http://192.168.50.13:6080/EmpresaB_1/services/getProductDetails

El tiempo de entrega del producto B0000205YX es : 1 semana en la tienda : 2
El precio encontrado del producto B0000205YX es : 549.99 en la tienda : 2
La cantidad encontrada del producto B0000205YX es : 233 en la tienda : 2
El * Id de Producto * es : B00005NTGG
El url es: http://192.168.50.13:6080/EmpresaB_1/services/getProductDetails

El tiempo de entrega del producto B00005NTGG es : 1semana en la tienda : 2
El precio encontrado del producto B00005NTGG es : 49.00 en la tienda : 2
La cantidad encontrada del producto B00005NTGG es : 158 en la tienda : 2
Finalizo la busqueda en la tienda. 2
*****
*****
Comienza la b'squeda en la tienda 1

Se hizo el hilo. Thread[Thread-34,5,main] a la tienda 1
El * Id de Producto * es : 47
El url es: http://192.168.50.13:6080/EmpresaA_1/services/getProductDetails

El * Id de Producto * es : 431718
El url es: http://192.168.50.13:6080/EmpresaA_1/services/getProductDetails

El tiempo de entrega del producto 431718 es : 1 semana en la tienda : 1
El precio encontrado del producto 431718 es : 60 en la tienda : 1
La cantidad encontrada del producto 431718 es : 962 en la tienda : 1
El * Id de Producto * es : 2
El url es: http://192.168.50.13:6080/EmpresaA_1/services/getProductDetails

```

Figura 3.8: Envío de solicitudes y resultados entre STD y las tiendas

resultados reflejan los productos encontrados en las 3 tiendas. Como se puede observar en la figura 3.2.6, el producto con número de identificador 47 no aparece en los resultados, esto indica al cliente que dicho producto no fue localizado en ninguna de las tiendas que seleccionó.

Los resultados son presentados en forma de lista, la primera columna indica la tienda donde se encuentra el producto. La columna identificador hace referencia al número de código único del producto. El precio del producto en la tienda está situado en la tercera columna. La columna de tiempo de entrega indica el tiempo en que el cliente tardará en recibir el producto.

El cliente utiliza la última columna en caso de que decida agregar a su carrito de compra alguno de los productos de la lista, si fuera el caso el cliente debe seleccionar el

Resultado de la Búsqueda - Microsoft Internet Explorer

Dirección: http://192.168.50.13:8080/std/Resultado.jsp

SURTETUDEPENSA

Tiendas Productos Pedidos Carrito Regístrate Ayuda

RESULTADO DE LA BÚSQUEDA

Se encontraron los siguientes productos:

Tienda	Identificador	Precio	Tiempo de Entrega	Comprar en esta tienda
Empresa A	431718	60	1 semana	<input type="checkbox"/>
Empresa A	B0000205YX	549	1 semana	<input type="checkbox"/>
Empresa A	B00005NTGG	49	1 semana	<input type="checkbox"/>
Empresa B	B0000205YX	549	1 semana	<input type="checkbox"/>
Empresa B	B00005NTGG	49	1 semana	<input type="checkbox"/>
Empresa B	431718	60	2 semanas	<input type="checkbox"/>
Gigante STD	431718	100	2 semanas	<input type="checkbox"/>
Gigante STD	2	90	1 semana	<input type="checkbox"/>
Gigante STD	19	78	1 semana	<input type="checkbox"/>

Aceptar

Ordenar en base a menor precio
 Ordenar en base a menor tiempo de entrega

Listo Intranet local

Figura 3.9: Resultados de la búsqueda

recuadro de la fila donde se encuentra el producto deseado seguido de presionar el botón de *Aceptar*.

En el momento en que el cliente agrega un producto al carrito, puede ver en la misma página de resultados una relación de los productos que contiene el carrito de compra en ese momento. En la parte izquierda de la figura 3.2.6 aparecen los resultados de la búsqueda, en la parte derecha se presentan los productos que el cliente agregó al carrito de compra. En esta última parte, se observa que la cantidad *10* contenida en el producto con número de identificador *B00005NTGG* está marcada con rojo, esto le indica al cliente que la cantidad de productos que solicitó es mayor a la existente en la tienda. De esta manera, la tienda solo podrá distribuir como máximo la cantidad de productos indicados en la columna de cantidad.

Por otra parte, la página de resultados ofrece la opción de listar los resultados en base a dos criterios, menor precio y menor tiempo de entrega. Menor precio, ordenará los resultados comenzando con el producto de menor precio sin importar la tienda donde

se localice. Menor tiempo de entrega, detallará los resultados iniciando con el de menor tiempo de entrega. En la figura 3.2.6 los resultados del cliente están ordenados en base al menor tiempo de entrega.

RESULTADO DE LA BÚSQUEDA

Se encontraron los siguientes productos:

Tienda	Identificador	Precio	Tiempo de Entrega	Comprar en esta tienda
Empresa B	B0000205YX	549.99	1 semana	<input type="checkbox"/>
Empresa A	431718	60	1 semana	<input type="checkbox"/>
Empresa A	B0000205YX	549.99	1 semana	<input type="checkbox"/>
Empresa A	B00005NTGG	49.00	1 semana	<input type="checkbox"/>
Gigante STD 2		90	1 semana	<input type="checkbox"/>
Gigante STD 19		78	1 semana	<input type="checkbox"/>
Empresa B	B00005NTGG	49.00	1 semana	<input type="checkbox"/>
Gigante STD 431718		100	2 semanas	<input type="checkbox"/>
Empresa B	431718	60	2 semanas	<input type="checkbox"/>

Ordenar en base a menor precio
 Ordenar en base a menor tiempo de entrega

PRODUCTOS EN EL CARRITO

Para modificar las cantidades que previamente había solicitado ingresar la nueva cantidad en el cuadro de texto correspondiente y después pulse el botón de Actualizar.

Tienda	Identificador	Cantidad	Precio	Eliminar
Empresa A	431718	1	60	<input type="checkbox"/>
Empresa A	B00005NTGG	10	49.00	<input type="checkbox"/>
Empresa B	B0000205YX	1	549.99	<input type="checkbox"/>
Gigante STD	2	1	90	<input type="checkbox"/>
Gigante STD	19	1	78	<input type="checkbox"/>

NOTA: Las cantidades marcadas con rojo indican que la cantidad solicitada por el usuario excede la cantidad en existencia que es la marcada por el número de color rojo.

Figura 3.10: Resultados de la búsqueda y productos agregados al carrito

3.2.7. Carrito de compra

Previo a su compra, el cliente elige de los resultados de la búsqueda los productos que contendrá su carrito de compra. Para ver la totalidad de productos existentes en el carrito, el cliente entra al enlace *Carrito*. En la figura 3.2.7 se puede visualizar todos los productos que el cliente agregó a su carrito. Los datos de los productos de carrito se encuentran detallados indicando tienda, producto, cantidad y tiempo de entrega de cada producto del carrito. La página de carrito contiene además de la lista de productos del carrito, la suma total de los productos. El cliente puede ver el monto total del pago.

En la lista de productos del carrito, las cantidades que exceden las existencias en las tiendas tendrán el formato explicado en la sección anterior. En la figura 3.2.7 se

observa que el producto marcado con el número de identificador *B00005NTGG* supera la cantidad en existencia, por esta razón es de color rojo. Una vez que el cliente está seguro de los productos que va a comprar y del total de la compra, presiona el botón de comprar finalizando con ello la compra.

SURTETUDESPENSA

Tiendas Productos Pedidos Carrito Regístrate Ayuda

CARRITO DE COMPRAS

Si desea modificar las cantidades que previamente había solicitado modifique la caja de texto correspondiente ingresando la nueva cantidad y después presione el botón de actualizar.

Tienda	Identificador	Tiempo de Entrega	Cantidad	Precio	Eliminar
Empresa A	431718	1 semana	1	60	<input type="checkbox"/>
Empresa A	B00005NTGG	1 semana	10	49.00	<input type="checkbox"/>
Empresa B	80000205YX	1 semana	1	549.99	<input type="checkbox"/>
Gigante STD	2	1 semana	1	90	<input type="checkbox"/>
Gigante STD	19	1 semana	1	78	<input type="checkbox"/>

Eliminar Actualizar

NOTA: Las cantidades marcadas con rojo indican que la cantidad solicita por el usuario excede la cantidad en existencia que es la marcada por el número de color rojo.

Subtotal	1512.99
Iva	226.0
TOTAL DE LA COMPRA	1738.99

Comprar

Figura 3.11: Carrito de compra

3.2.8. Ayuda

El último botón de enlace que contiene el sistema es el de *Ayuda*. Si el cliente está utilizando el sistema por vez primera y no conoce la forma correcta de manejarlo, puede referirse a la página de ayuda al escoger el enlace de *Ayuda*. Esta página describe de modo muy general el funcionamiento del sistema y la forma de operarlo. Además, se pueden encontrar en esta página una lista de las personas participantes en el desarrollo del proyecto.

En la figura 3.2.8 se nota el contenido de la página de ayuda. La página explica brevemente cual es el objetivo de STD, su importancia y su funcionamiento. Al final de

la página se enumeran los pasos que deben ejecutarse para utilizar el sistema de forma adecuada.

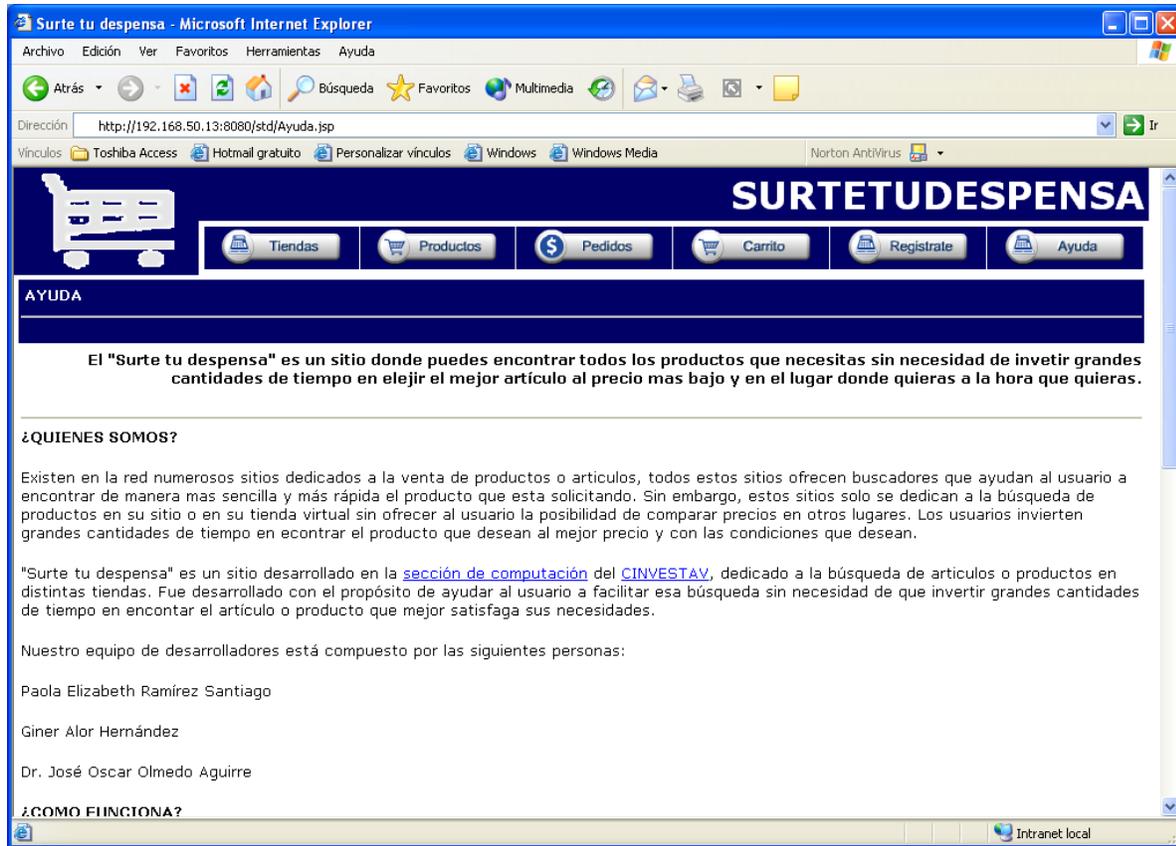


Figura 3.12: Ayuda del sistema

3.3. Resumen

La mayor parte de los portales dedicados a la venta de productos que se encuentran en Internet, promocionan exclusivamente productos de su empresa. Las personas que necesitan adquirir un producto, buscan las mejores alternativas de precio invirtiendo grandes cantidades de tiempo en encontrar la tienda que ofrezca la mejor opción. STD ayuda a resolver esta tarea, ofreciendo un sistema que permite buscar las mejores opciones del producto deseado en un solo lugar. Con el único fin de probar la utilidad de este sistema, en este capítulo se expuso un caso de estudio.

El caso de estudio descrito mostró una de las formas en las que se puede aplicar el sistema en forma práctica. La búsqueda de los productos, se realizó en tres tiendas registradas previamente. El cliente escogió las tiendas y el orden de búsqueda en ellas. El resultado de la búsqueda arrojó las opciones solicitadas por cliente en las tiendas de su preferencia. El cliente finalmente eligió los productos a comprar en base a los resultados obtenidos de la búsqueda, ahorrando con ello tiempo de búsqueda y obteniendo las mejores opciones de productos en las tiendas de su preferencia.

Capítulo 4

Diseño del sistema STD

En el presente capítulo se describe el diseño del sistema STD. El sistema STD se analiza desde cuatro enfoques diferentes: organizacional, funcional, el modelo de datos y el modelo de comportamiento. El enfoque organizacional describe los componentes en los que se encuentra constituido STD. Se explica el papel que desempeña cada uno de estos así como sus responsabilidades. El enfoque funcional expone las diferentes operaciones que ofrece el sistema. Se describen las salidas que producen los servicios como respuesta a las peticiones que reciben y los datos que se generan en la respuesta.

El enfoque del modelo de datos detalla la representación de los datos que son utilizados en el sistema. Se visualiza el contenido de las diferentes bases de datos creadas para el funcionamiento correcto del sistema, además de las relaciones existentes entre las mismas. Finalmente, en el enfoque del modelo de comportamiento se describe el procedimiento que desarrollan las operaciones descritas en el modelo funcional.

4.1. Descripción organizacional

En el capítulo anterior describimos el funcionamiento general del sistema con ayuda de un caso de estudio, en base a esto podemos señalar que los componentes del sistema son esencialmente tres: el cliente, STD y los servidores comerciales como se muestra en la figura 4.1.

Como se puede observar en la figura 4.1, STD está integrado por 3 componentes que interactúan entre sí con el objeto de responder a las solicitudes elaboradas por los clientes. En STD los componentes se encuentran organizados siguiendo el esquema cliente-servidor.

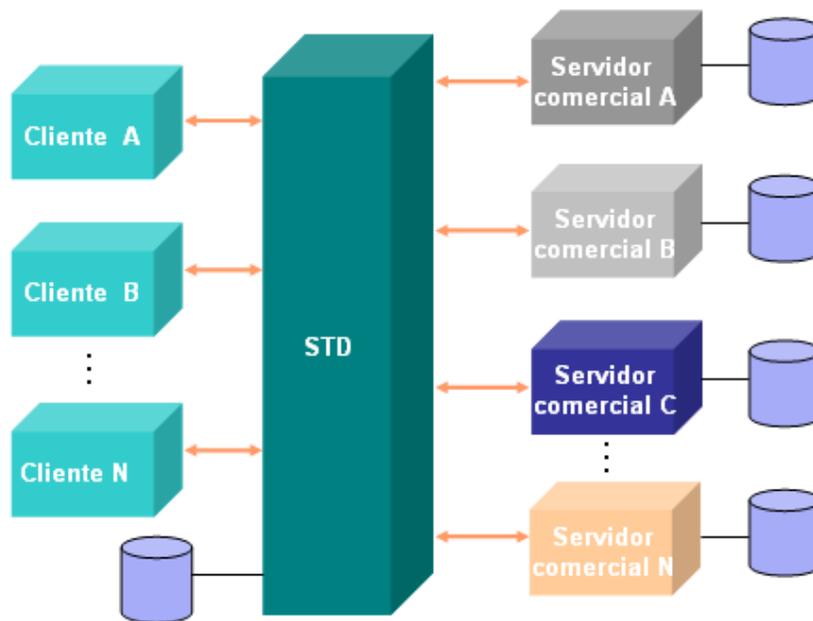


Figura 4.1: Diseño arquitectural de STD

- a. **El cliente:** Podemos definir a un cliente como cualquier persona que hace uso de un navegador de Internet. El cliente es responsable de iniciar la interacción con el sistema al iniciar la operación comercial solicitando un pedido ó simplemente al hacer uso de algunos de los servicios que ofrece STD. Así mismo, debe informar a STD en caso de presentarse alguna anomalía en los datos que tiene registrados ó si desea actualizarlos. El cliente puede acceder desde cualquier navegador al sistema, sin embargo, debe estar registrado como cliente del sistema para que pueda hacer uso de los servicios que ofrece.
- b. **STD:** STD actúa como servidor de comercio electrónico. Este servidor de comercio electrónico se encuentra situado en el servidor Web. El servidor Web donde fue implantado STD es Tomcat en su versión 5.0.9.

STD es responsable de satisfacer las solicitudes hechas por los clientes, para ello utiliza la tecnología JSP (Java Server Pages). En las páginas JSP que se encuentran en el servidor, se localiza el código Java responsable de responder a la solicitud del cliente. Además, el servidor hace uso de las diferentes bases de datos donde se encuentra la información solicitada por el cliente. Finalmente, el cliente recibe la respuesta a su solicitud en su navegador ver figura 4.2.

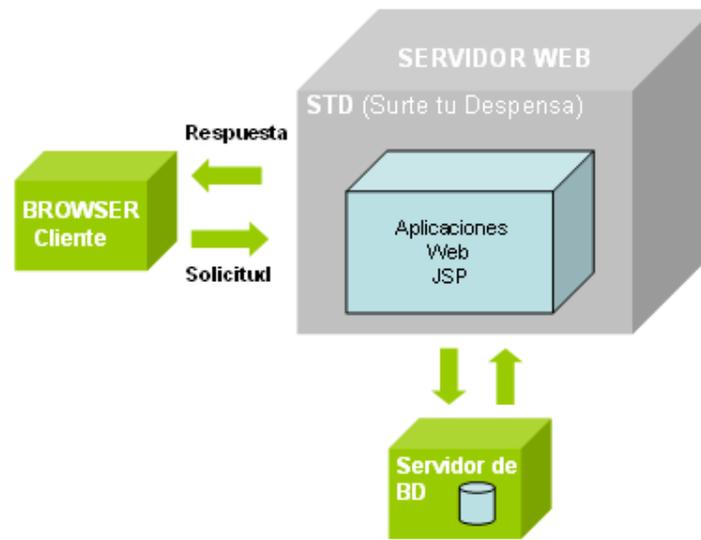


Figura 4.2: Interacción cliente – servidor Web

- c. **El servidor comercial:** Todas aquellas empresas que se dediquen a la venta de productos y que se encuentren registradas dentro de STD actúan como servidores comerciales. El servidor comercial se conduce como proveedor de productos mismos que son ofrecidos en STD.

El servidor comercial podrá contar con un Portal que ofrezca la venta de sus productos en línea. Sin embargo, el requisito indispensable para que los productos de este servidor puedan ser mostrados dentro de STD, es que el servidor desarrolle los servicios Web necesarios para obtener la información de los productos que ofrece. De esta manera, los productos son expuestos para su venta en STD.

El servidor comercial publica sus servicios Web, los cuales se registran en STD para poder llevar a cabo la interacción entre ellos. En STD pueden existir varios servidores comerciales. En el desarrollo y las pruebas del sistema STD se utilizaron 3 servidores comerciales.

El servidor comercial es responsable de crear los servicios Web necesarios para obtener la información de los productos que vende. Asimismo, debe publicar los servicios Web que desarrolló para que puedan ser invocados desde STD. También, es responsable de actualizar la información de los productos que ofrece y que se encuentran dentro de las Bases de Datos de su empresa.

4.2. Descripción funcional

STD tiene como principal objetivo el de satisfacer listas de pedidos definidas por los clientes, obteniendo los resultados tanto como sea posible a los deseados por estos. Cada uno de los componentes que intervienen para el logro de estos resultados, tiene a su cargo diferentes operaciones en las cuales se basa para lograr el objetivo deseado.

El Cliente es el único de los componentes que no tiene a su cargo ninguna operación, simplemente está dedicado al envío de peticiones que forma mediante su navegación en el Portal. STD es el principal encargado de realizar las operaciones necesarias para poder responder a las peticiones del cliente, auxiliándose de las operaciones que ofrecen los sitios comerciales.

4.2.1. Funcionalidad de STD

Como ya se mencionó en la sección 4.1, STD actúa como servidor de comercio electrónico y es el encomendado para responder a las peticiones de los clientes. Para lograr esto STD se auxilia de varios servicios. Los servicios que ofrece STD son los siguientes:

- Registro de clientes
- Administración de listas de productos
- Administración de listas de tiendas
- Administración de listas de clientes
- Administración de pedidos
- Búsqueda de productos
- Visualización del resultado del pedido
- Actualización de datos
- Administración del carrito de compras

Para que el usuario pueda hacer uso de estos servicios, tiene que solicitar la operación mediante una sesión con el servidor a través del navegador. Muchos de los servicios que ofrece STD solicitan información de entrada que necesitan para procesar la petición. Esta información, se proporciona por los clientes que se encargan de llenar los formularios

correspondientes. Si es el caso, STD obtiene la información de las ligas o hipervínculos que visite el cliente. La salida o respuesta generada desde STD, se visualiza por el cliente mediante una página Web, la cual puede ser vista desde su navegador. Algunas de las respuestas generadas se ven reflejadas a través de actualizaciones en las bases de datos, como es el caso de la modificación ó actualización de los datos del cliente.

Registro de clientes

Todo usuario que persiga utilizar STD debe estar registrado en el sistema. El registro se lleva a cabo en el momento en que el usuario llena un formulario en el que introduce los datos solicitados por el sistema. STD guarda su información en la base de datos. Como respuesta a la operación de registro solicitada por el cliente, se envía a su navegador una página Web que le indica si el registro fue exitoso o en su defecto si no lo fue.

Administración de listas de tiendas

Como se mencionó en la sección 4.1, para el desarrollo del sistema STD se crearon 3 tiendas que actúan como servidores comerciales. Estos servidores comerciales están registrados en STD al igual que los servicios Web que proporcionan.

STD administra a las empresas que desean ofrecer sus productos en el sistema realizando su registro y actualizando sus datos siempre que así lo requieran. Además de ello, registra los servicios Web que le proporciona cada tienda con el objeto de lograr su posterior invocación.

Administración de listas de clientes

Todos los registros de clientes se almacenan en la base de datos. STD actualiza los datos de los clientes que así lo soliciten o en el caso de presentarse alguna modificación en los mismos. Los datos que STD guarda respecto de cada uno de los clientes son aquellos dados al momento de su registro.

Administración de listas de productos

STD pone a disponibilidad de sus clientes la lista de los productos que ofrece. Los productos que los clientes encontrarán en esta lista son aquellos que pueden agregar a su pedido para la búsqueda en las tiendas. Debido a la gran diversidad de productos, esta lista se encuentra almacenada en una base de datos y se agrupa por semejanzas entre ellos. Es decir, la lista de productos que ofrece STD está clasificada por categorías. STD

cuenta con 24 categorías. De igual manera, la mayor parte de las categorías contiene una o más subcategorías en las que se puede observar más específicamente el tipo de productos que la componen.

STD actualiza los productos correspondientes a cada categoría en caso de que se necesite y de igual forma modifica sus subcategorías. La cantidad de subcategorías procedente de cada categoría depende del tipo de categoría.

Administración de pedidos

Uno de los servicios que provee STD es el de creación de pedidos. Los clientes registrados en el sistema pueden crear su pedido y solicitarlo en búsqueda. El cliente inicia la elaboración del pedido en el período en que escoge los productos que contendrá. Puede navegar entre las categorías y las subcategorías hasta encontrar el producto de su preferencia. El cliente solicita al sistema la cantidad de unidades por producto que necesite; de no ser así, el sistema toma la cantidad predeterminada que es de una unidad.

En el momento en el que el cliente termina de escoger los productos de su preferencia, su pedido está listo para iniciar la búsqueda. La lista de productos que forma el pedido, puede verse desde el navegador del cliente. El cliente puede actualizar ó eliminar los productos de su pedido en caso de requerirlo. También puede aumentar o reducir la cantidad de unidades correspondientes a cada producto antes de finalizar el pedido.

Búsqueda de productos

En la sección anterior se describió el servicio que permite crear y administrar un pedido. El servicio de búsqueda de productos toma el pedido creado previamente antes de iniciar la búsqueda.

STD necesita conocer además del pedido, la lista de tiendas en las que el cliente espera llevar a cabo la búsqueda. STD muestra al cliente un listado de las tiendas (servidores comerciales) en las que puede realizar la búsqueda. El cliente escoge el orden de prioridad de la búsqueda, es decir, la tienda que tenga el número uno es la primera tienda en la que se busca el pedido o la tienda de mayor prioridad. La tienda que tenga el número más grande será la de menor prioridad. Una vez que tenga el pedido y el orden de prioridad de las tiendas, el sistema podrá iniciar la búsqueda de los productos.

Visualización del resultado pedido

Al finalizar la búsqueda, STD muestra al cliente un listado de los resultados obtenidos en las diferentes tiendas. El cliente puede visualizar en su navegador una página en la que verá el resultado de los productos encontrados en las tiendas. STD publica al cliente una tabla en la que detalla el resultado del pedido. Este servicio permite al cliente visualizar el resultado ordenado de dos formas diferentes: de acuerdo al menor precio ó al menor tiempo de entrega. Para mayor información consultar la sección 3.2.6.

Actualización de datos

Como ya hemos visto, para que el cliente pueda obtener los productos que necesita debe realizar una serie de operaciones que concluyen con el resultado de su pedido. En algunas de las operaciones que ofrece STD se pueden actualizar datos, entendiéndose por este término el de modificar datos existentes registrados en el sistema por nuevos.

El cliente puede actualizar la lista de productos de su pedido cuando se requiera o bien eliminar un producto existente. También, puede modificar la cantidad de unidades solicitadas por dicho producto ya sea antes de solicitar su pedido, en el momento de obtener su resultado o antes de terminar su compra. STD actualiza de igual forma datos de productos, clientes y tiendas que así lo requieran, solicitando previamente esta operación al sistema.

Administración del carrito de compras

Una vez que el cliente haya escogido los productos que desea comprar de la lista del resultado de su búsqueda, estos productos pueden visualizarse en su carrito de compra. El servicio de carrito de compra proporciona al cliente la facilidad de revisar, previo a su compra, una lista de los productos que el eligió para comprar. En el carrito de compra se provee al cliente la siguiente información: tienda, identificador, tiempo de entrega, cantidad, precio y eliminar.

La información referente a tienda, identificador, tiempo de entrega y precio coincide con aquella que se mencionó en la sección 3.2.6. La cantidad especifica el número de unidades por cada producto que se desean comprar. La opción de eliminar permitirá al cliente el borrado de los productos que ya no desea comprar. El servicio del carrito de compras también permite que el cliente pueda conocer el total de su pedido. El subtotal de los productos resultará de multiplicar la cantidad de cada producto por su precio correspondiente, finalizando con la suma de todas las multiplicaciones. Al subtotal se le

sumará la cantidad correspondiente al IVA de la compra finalizando con el total que se pagará por los productos solicitados.

4.2.2. Funcionalidad de los servidores comerciales

Los servidores comerciales, tienen a su cargo el cumplimiento de varias tareas como se mencionó en la sección 4.1. Para lograr esto, se auxilian de varios servicios. Los servicios que ayudan a los servidores comerciales con el cumplimiento de sus tareas son:

- Creación y publicación de servicios Web
- Búsqueda de productos solicitados
- Formar lista de respuesta a la búsqueda solicitada
- Administración de listas de productos

STD se auxilia de las operaciones que ofrecen los servidores comerciales para responder a las solicitudes de los clientes. STD pide a los servidores comerciales, información referente a los productos que ofrecen invocando sus servicios Web. Los servidores responden a estas peticiones realizando las operaciones correspondientes dependiendo de cada petición. STD utiliza el resultado y lo procesa para mandar la respuesta final a la solicitud elaborada por el cliente. Para el envío de peticiones entre STD y los servidores comerciales y la respuesta por parte de estos últimos, se utiliza el protocolo SOAP.

Creación y publicación de servicios Web

Para que los servidores comerciales puedan ofrecer sus productos en STD, deben contar con los servicios Web que proporcionen la información necesaria para invocarlos. Cada servidor comercial puede crear diferentes servicios Web dependiendo de sus necesidades, pero para que sus servicios Web puedan ser invocados desde STD deben cumplir con determinadas condiciones. El servicio Web desarrollado por el servidor comercial debe pedir como parámetro de entrada el código de producto para poder desarrollar la búsqueda. Este código corresponde a un código único con el que se identificará a ese producto. El servicio Web enviará como respuesta a la solicitud de STD los siguientes datos: nombre del producto, cantidad, precio y tiempo de entrega.

En seguida de que los servicios Web fueron creados por el servidor comercial, deben publicarse para su posterior invocación desde STD. La invocación se lleva a cabo en el momento de la búsqueda de productos.

Búsqueda de productos solicitados

Una vez que STD ha invocado a los servicios Web que ofrecen los servidores comerciales, estos comenzarán con la búsqueda del producto solicitado. Como se mencionó en la sección anterior, el servidor comercial recibe como parámetro de entrada desde STD el código del producto a buscar. Esta búsqueda se lleva a cabo dentro de las Bases de Datos del servidor.

El servidor comercial busca en sus bases de datos los valores correspondientes a la siguiente información: nombre, cantidad, precio y tiempo de entrega del producto. Estos datos son dependientes del código de producto que se obtuvo de STD. El resultado de los valores buscados se envía a STD. STD procesa el resultado enviado por servidor comercial y lo utiliza de diversas formas presentando la información solicitada por el cliente.

Formar lista de respuesta de búsqueda solicitada

Una vez que el servidor termina de procesar la solicitud hecha por STD, el servidor le envía el resultado obtenido. El resultado refleja el valor de los datos del producto encontrados en el servidor. La información relativa al nombre, la cantidad y el precio del producto no deben generar valores nulos como respuesta a la solicitud. Sin embargo, el tiempo de entrega puede ser un valor nulo.

Administración de listas de productos

El servidor comercial posee una base de datos en la que almacena su lista de productos. Este servidor se encarga de actualizar sus Bases de Datos apoyándose en la venta de sus productos, ya sea directamente con su empresa o mediante STD. También debe actualizar las listas de precios de sus productos. Los datos que STD muestra al cliente son los que se encuentran en el servidor comercial al momento de requerirlos.

4.3. Descripción de los datos

Para su funcionamiento STD realiza varias operaciones, como se mencionó en las secciones anteriores. La mayor parte de estas operaciones requieren de datos que toman de STD para procesar peticiones. Asimismo, STD maneja una gran diversidad de productos que necesita manipular y actualizar constantemente. Para poder operar esta gran cantidad de datos, STD se auxilia de diversas tablas contenidas en una base de datos.

Para el desarrollo del sistema STD, se organizaron 4 bases de datos diferentes. La primera de ellas corresponde a la base de datos principal del sistema. En esta base de datos, STD deposita datos propios del sistema. Las bases de datos restantes contienen la información correspondiente a los productos que ofrecen cada uno de los servidores comerciales.

4.3.1. Base de datos principal de STD

STD utiliza una base de datos en la que almacena toda su información. La base de datos principal de STD cuenta con 9 tablas, cuyo esquema se muestra en la figura 4.3. Como se puede observar, la figura 4.3 presenta las tablas contenidas en la base de datos de STD y sus campos correspondientes. Las tablas de Categorías y Subcategorías se encuentran relacionadas por el campo IdCategoría, mientras que las tablas restantes no contienen relación alguna. Las tablas mostradas en la figura 4.3 se describen a continuación con mayor grado de detalle.

Tabla de Tiendas. Esta tabla contiene la información correspondiente a los servidores comerciales en los que realiza la búsqueda de productos de STD. Los campos de la tabla de Clientes se muestran en la figura 4.3. Cada tienda tiene un número único con el que se distingue de todas las demás; este número se guarda en el campo Idtienda. Todos los campos de la tabla son llenados con la información que proporciona la empresa que se registra, a excepción del campo Idtienda que es proporcionado por STD.

Tabla de Clientes. En la tabla de Clientes se guarda toda la información de aquellos clientes que se hallen registrados en el sistema. El campo IdCliente, es almacenado con la contraseña que proporcione el cliente al momento de su registro. En el resto de los campos se deposita la información que facilita el cliente al registrarse.

Tabla de Productos. En la tabla de Productos se registran todos y cada uno de los productos que están disponibles para su búsqueda en STD. Los datos que contiene esta tabla reflejan los diversos productos que ofrecen las tiendas registradas en STD. Cada producto contiene un código único que lo hace diferente de los demás, este código se guarda en el campo IdProducto. Este código debe coincidir con el que tienen registrado las diferentes tiendas. En el campo NombreProducto se precisa el tipo de producto al que se hace referencia con IdProducto.

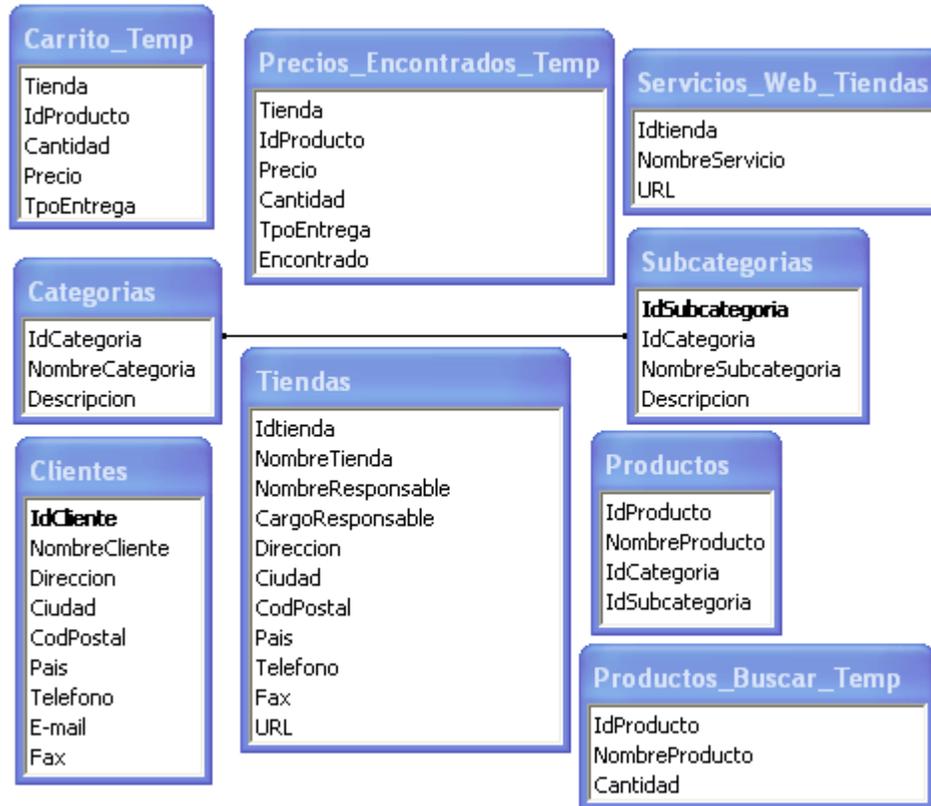


Figura 4.3: Tablas de las base de datos de STD y sus relaciones

Cada producto ha sido clasificado en una categoría y en algunas ocasiones dentro de una subcategoría dependiendo de su tipo. En los campos `IdCategoria` e `IdSubcategoria` se deposita el número de la categoría y subcategoría que corresponde el producto identificado con `Idproducto`.

Categorias. Para facilitar la búsqueda de los productos en STD se organizaron 24 categorías, mismas que se hallan descritas en esta tabla. A cada categoría se le designó un número único que se refleja en el campo `IdCategoria`. En los campos `NombreCategoria` y `Descripción`, se guarda la información correspondiente al nombre y la descripción de la categoría propia de `IdCategoria`.

Subcategorias. La mayor parte de las categorías tienen una segunda clasificación en subcategorías. En la tabla de Subcategorías se almacenaron los datos de cada una de las subcategorías. STD cuenta con 121 subcategorías. El campo `IdSubcategoria` designa un número único de subcategoría. El campo `IdCategoria` que se localiza en esta tabla

hace referencia al número de categoría al que pertenece la subcategoría. En los campos restantes se define el nombre y la descripción de la subcategoría.

Servicios_Web_Tiendas. En esta tabla se guarda la información necesaria para invocar a los servicios Web de las tiendas registradas en STD. El campo Idtienda hace referencia a la tienda a la que corresponde el servicio Web. El nombre del servicio se especifica en el campo NombreServicio. El campo URL contiene la dirección en donde se hace la invocación del servicio Web.

Productos_Buscar_Temp. En esta tabla se guarda un listado de los productos que el cliente eligió para su búsqueda dentro de STD. En los campos Idproducto y NombreProducto se recolecta la información del producto ya sea por su número identificador o por su nombre. El campo Cantidad contiene el total de unidades por producto que se solicitan para la búsqueda del mismo.

Precios_Encontrados_Temp. La tabla de Precios_Encontrados_Temp deposita la información que se obtuvo como resultado de la búsqueda del pedido solicitado por el cliente en las diferentes tiendas. El campo Tienda contiene el número identificador de la tienda en la que se encontró el producto, mismo que se describe en el campo Idproducto. Los campos Precio, Cantidad y TpoEntrega almacenan la información que se obtuvo del producto. Si el producto se encontró en la tienda, el campo Encontrado tendrá el valor de uno, en caso contrario el número será cero.

Carrito_Temp. En esta tabla se guarda la información de los productos que el cliente eligió para su compra. Los campos de esta tabla son equivalentes a los de la tabla Precios_Encontrados_Temp, a excepción del campo Encontrado que se omite en esta tabla.

4.3.2. Bases de datos de los servidores comerciales

Cada servidor comercial simula una empresa ficticia. Asimismo, a cada uno se le asignó un nombre diferente. La primera empresa fue denominada como Empresa A, la segunda como Empresa B y la tercera lleva el nombre de Gigante STD. Se diseñó una base de datos diferente para cada empresa.

a) Bases de datos de las empresas A y B

Las empresas A y B poseen bases de datos equivalentes, las cuales fueron organizadas con el apoyo de la estructura de las bases de datos contenidas en la empresa *Amazon* (<http://www.amazon.com>). Es decir, el diseño que poseen las empresas A y B se asemeja en gran parte al diseño que tiene Amazon. Para lograr esto, se desarrolló un programa en Java que mediante la invocación de los servicios Web de Amazon, obtuvo datos reales de los productos que ofrece. Se discute más a fondo este tema en el capítulo 5.

Los datos obtenidos del programa se almacenaron en las bases de datos de las empresas A y B. La base de datos de las empresas A y B tiene 14 tablas, 12 de ellas se apoyan en el diseño de Amazon. Las 2 restantes que corresponden a Pedidos y Clientes, se crearon en base a los requerimientos básicos de las empresas. El esquema de las bases de datos de las empresas A y B se muestra en la figura 4.4.

Las tablas no contienen ningún tipo de relación. Pese a que cada empresa cuenta con 14 tablas, STD solo maneja la información de una de ellas. La tabla de Productos es la única de todas las tablas en la que STD busca información de los productos solicitados por los clientes. Esto debido a que es en ella donde se almacena la información principal de los productos con los que cuenta la empresa.

El resto de las tablas exceptuando las tablas Pedidos y Clientes, fueron diseñadas con el objeto de que la empresa guardara la información de los productos que ofrece con alto grado de detalle. La mayor parte de las tablas contenidas en la figura 4.4 contienen el campo Asin. Este campo identifica al producto con un número único. De esta manera, podemos identificar los datos del producto en cada una de las tablas.

Podemos ejemplificar esto de la siguiente manera, la Empresa A tiene registrado en su base de datos un producto que contienen como Asin el número AFXCTY6. En la tabla de Productos encontraremos las características principales de este producto. El campo de ProductName de la misma tabla indica que el producto es un disco compacto de música. En la tabla de Tracks se localizarán los títulos de las canciones contenidas en el disco compacto.

La Tabla de Pedido que se muestra en la figura 4.5, tiene el objetivo de guardar los datos de los pedidos solicitados a la empresa. Cada pedido cuenta con un número identificador designado en el campo IdPedido, el identificador del producto que se encuentra en el pedido se almacena en el campo Idproducto. La Cantidad corresponde al total de unidades del producto que contiene el pedido, finalmente en el campo IdCliente se registra el número identificador del cliente quien realizó el pedido.



Figura 4.4: Tablas de las base de datos de las empresas A y B

La tabla de Clientes es igual a la tabla que se lleva el mismo nombre en la bases de datos de STD, para mayor información revisar la sección 4.3.1.

b) Base de datos de Gigante STD

La Empresa Gigante STD fue estructurada de manera muy distinta a las anteriores. Su diseño está basado en el tipo de organización que presentan algunas de las empresas que actualmente disponen de un Portal Web en el que venden productos y/o servicios en línea. Estas empresas son: *Comercial Mexicana*, *Gigante*, *Superama*, *Liverpool*, *Palacio de Hierro* y *Home Mart México*. La base de datos de Gigante STD posee solo 3 tablas, su esquema su observa en la figura 4.6.

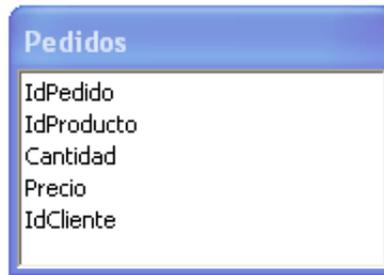


Figura 4.5: Tablas de Pedidos de las empresas A y B

La empresa Gigante STD organiza su información de una manera similar a STD. Los productos están constituidos de tal forma que cada uno tiene un identificador único.

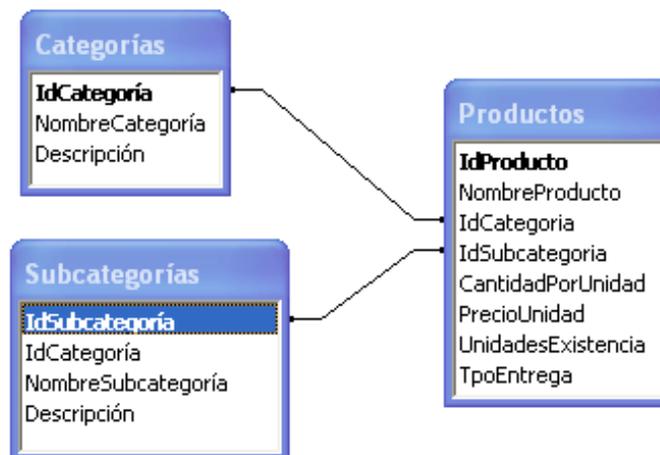


Figura 4.6: Tabla de la base de datos de la empresa GiganteSTD

Tabla de Productos. La tabla de productos guarda el registro de todos los productos existentes en la tienda. Debido a que cada producto se encuentra clasificado en una categoría y en una subcategoría, los campos IdCategoría e IdSubcategoría indican respectivamente el número correspondiente a ellas. El campo de CantidadPorUnidad muestra el tipo y la cantidad de unidades que se maneja para cada producto (lt, kg, sobres, etc.).

El campo de Precio expresa el precio del producto. El campo UnidadesExistencia revela la totalidad de productos que existen en almacén. Finalmente, en el campo TpoEntrega podremos ver el tiempo que Gigante STD tarda en hacer entregar del producto, si es que el cliente finalmente decide comprarlo. La tabla de Clientes se conduce de igual

forma que la mencionada en el punto 4.3.1.

4.4. Descripción del comportamiento

STD, se desarrolla en un entorno en el que interactúa tanto con clientes como con servidores comerciales. Asimismo, cada una de estas partes realiza varias operaciones con las que llevan a cabo esta interacción. En la figura 4.7 se muestra el diagrama de casos de uso del sistema STD. En el se puede percibir en términos generales la funcionalidad que ofrece STD. En la figura podemos ver todas las actividades que realiza STD y su interacción con el entorno.

El sistema interactúa con 2 actores, el cliente y los servidores comerciales. Estos actores desempeñan un papel externo en el sistema que realiza algún tipo de interacción con el mismo. El Cliente es quien va a hacer uso de las diferentes operaciones que ofrece STD (representados como casos de uso). Cada objetivo o propósito del usuario se representa como un caso de uso. Algunas de estas actividades se van a auxiliar de otras para procesar sus peticiones. STD va a interactuar con los servidores comerciales registrados, representados en el diagrama como Empresa A, Empresa B y Gigante STD solicitándoles la búsqueda de los productos.

Hasta este momento, hemos definido cuáles son las actividades que lleva a cabo el sistema pero no el cómo se realizan. Cada una de las actividades u operaciones describe una secuencia de interacciones que se producen entre un actor y el sistema cuando el actor usa el sistema para llevar a cabo una tarea específica. El nombre de cada caso de uso refleja la tarea que el actor desea llevar a cabo usando el sistema.

En esta sección, representaremos el ordenamiento que debe darse a las tareas que se desarrollan en una operación o servicio para satisfacer las condiciones de los datos de entrada y de salida dados en la descripción funcional. El comportamiento que sigue cada una de las actividades que contiene STD se explica a continuación detalladamente.

Registro de Clientes. Para poder registrar un cliente se realizan las siguientes tareas. Inicialmente el usuario debe ingresar al servicio de registro. El usuario envía al sistema los datos solicitados por el mismo, entonces STD comprueba que los datos suministrados por el cliente sean válidos.

Se almacena en la base de datos un nuevo registro que contiene los datos válidos del cliente. Al finalizar, el cliente recibe un mensaje desde el sistema indicando el estado de su registro, que podría ser como se muestra en la figura 4.4, un registro terminado

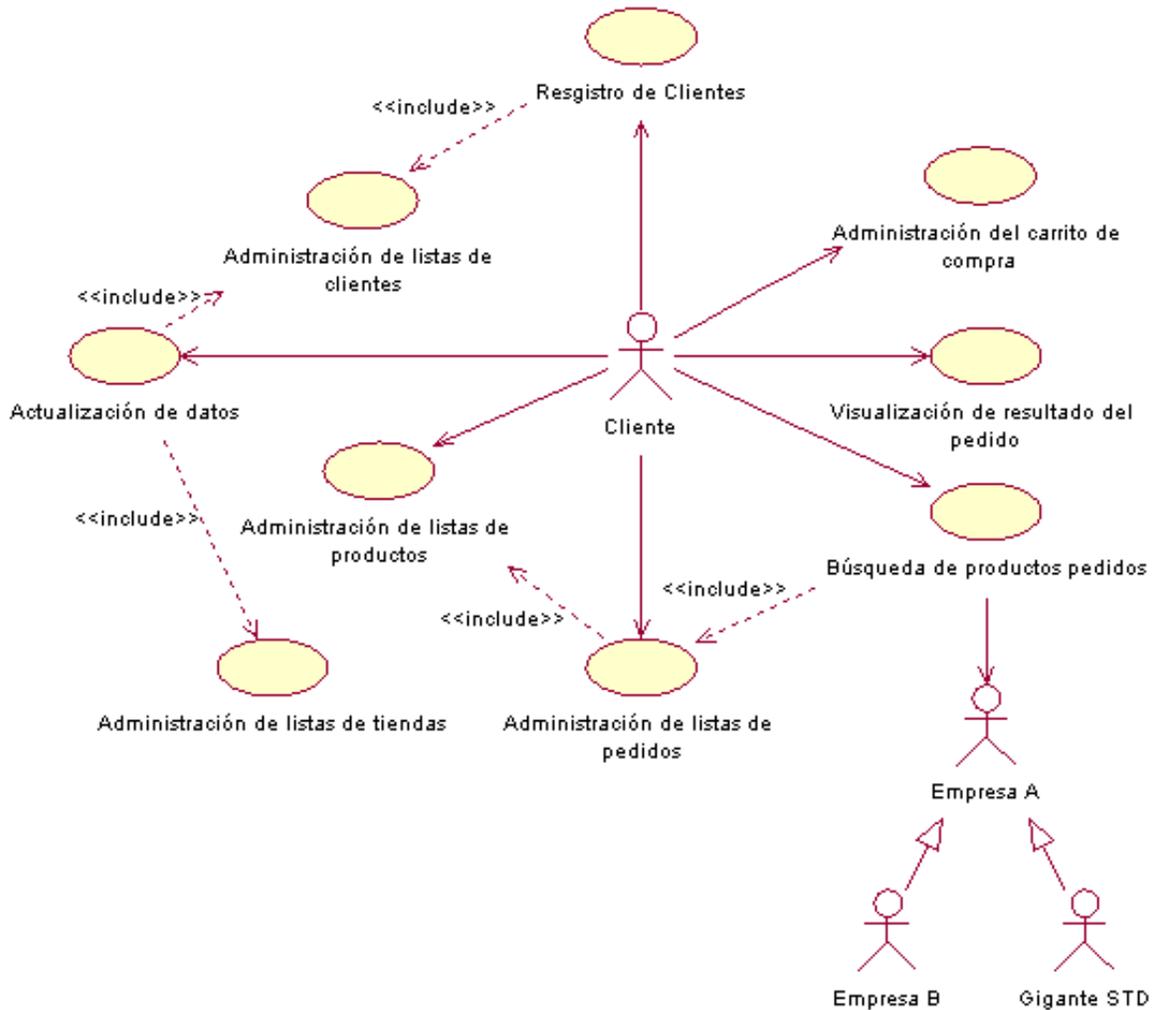


Figura 4.7: Diagrama de Casos de uso de STD

exitosamente ó un mensaje de error.

Administración de listas de clientes. Al realizar esta operación el cliente desarrolla las siguientes tareas. Primeramente el cliente solicita al sistema que actualice algunos de sus datos guardados previamente. El sistema mediante una consulta SQL busca que el cliente se encuentre dentro de la base de datos.

Si el cliente está registrado en el sistema, este último confirma que los nuevos datos aportados por el cliente sean correctos. Una vez que STD verifique la validez en los datos, los actualiza en la base de datos; si el registro finalizó exitosamente, el usuario recibe un mensaje en el que se le indica el estado del registro (ver figura 4.4.)

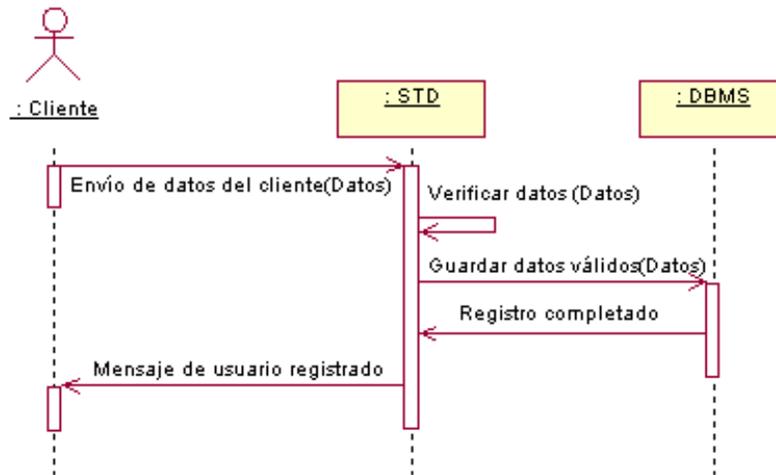


Figura 4.8: Diagrama de secuencia del registro de clientes

Administración de listas de tiendas. Este servicio cumple con dos tareas principales: el registro y la actualización de tiendas. La figura 4.4 representa el comportamiento que sigue el sistema al registrar una tienda.

Toda tienda que desee ofrecer sus productos en STD debe registrarse como se ha mencionado anteriormente. Para ello, remite a STD una solicitud de registro de tienda. STD analiza de ser posible, la solicitud de registro tomando en cuenta las características propias de la tienda. Una vez que se aceptó la solicitud de registro, la tienda facilita a STD los datos necesarios para completar el registro. Los datos se revisan y al final de su revisión en caso de ser correctos se guardan en la base de datos correspondiente. Después de que se depositaron los datos de la tienda, esta debe proporcionar a STD los servicios Web necesarios para obtener la información de los productos que va a ofrecer en STD.

STD verifica que los servicios Web estén especificados de manera correcta. Los servicios Web se almacenan en la base de datos para su posterior invocación. En último lugar, la tienda es informada sobre el registro. Además de registrar una tienda, la operación de administración de listas de tiendas consentirá que las tiendas actualicen los datos que STD almacenó al momento de su registro. Las actualizaciones pueden surgir en los siguientes casos: la tienda necesita modificar alguno de sus datos registrados debido a que cambiaron por alguna razón especificada. La tienda modifica sus datos porque se identificó un error en ellos.

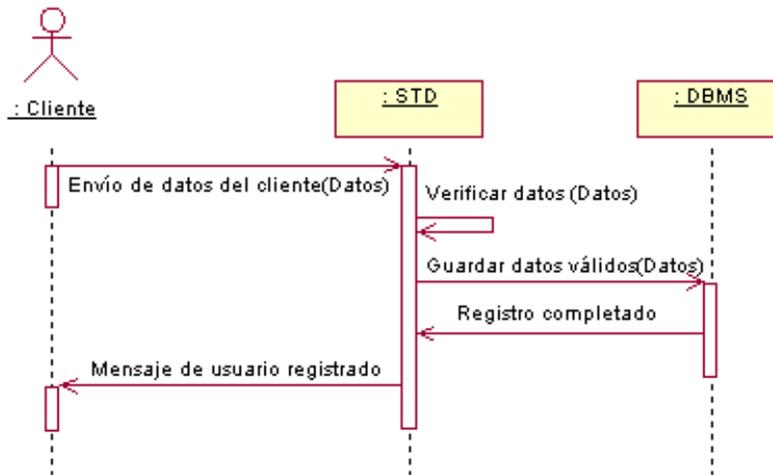


Figura 4.9: Diagrama de secuencia de administración de listas de clientes

La figura 4.4 representa el procedimiento que se realiza en el momento en que una tienda actualiza su información. El servidor comercial solicita a STD la renovación de sus datos, requiriendo precisar las razones por las cuales desea llevar a cabo la modificación. STD busca el registro de la tienda que solicita la actualización dentro de la base de datos. En caso de que la tienda exista, los datos se actualizarán y el registro será completado. STD debe informar a la tienda en caso de finalizar con éxito la actualización, o si esto no ocurriera, las razones por las cuales no se pudo concluir la operación.

Administración de listas de productos. El cliente puede visualizar los productos que se localizan en STD y elegir los de su preferencia en la operación de administración de listas de productos. Para completar la operación se realiza el siguiente procedimiento. El cliente pide a STD que le facilite la lista de productos que ofrece, STD busca los productos en la base de datos y envía un listado de ellos agrupados por categorías. El cliente puede observar esta lista desde su navegador.

Después de que el cliente revise todas las categorías, pide a STD la lista de los productos de una determinada categoría. STD analiza si esa categoría contiene subcategorías. Si la categoría elegida contiene subcategorías, STD proporciona al cliente un listado de las subcategorías que contiene la categoría elegida previamente. El cliente escoge la subcategoría de la que desea ver los productos y envía una solicitud a STD para que lleve a cabo la búsqueda. Al finalizar la búsqueda en la base de datos, STD da al cliente una lista de los productos que contiene la subcategoría que escogió. Si la

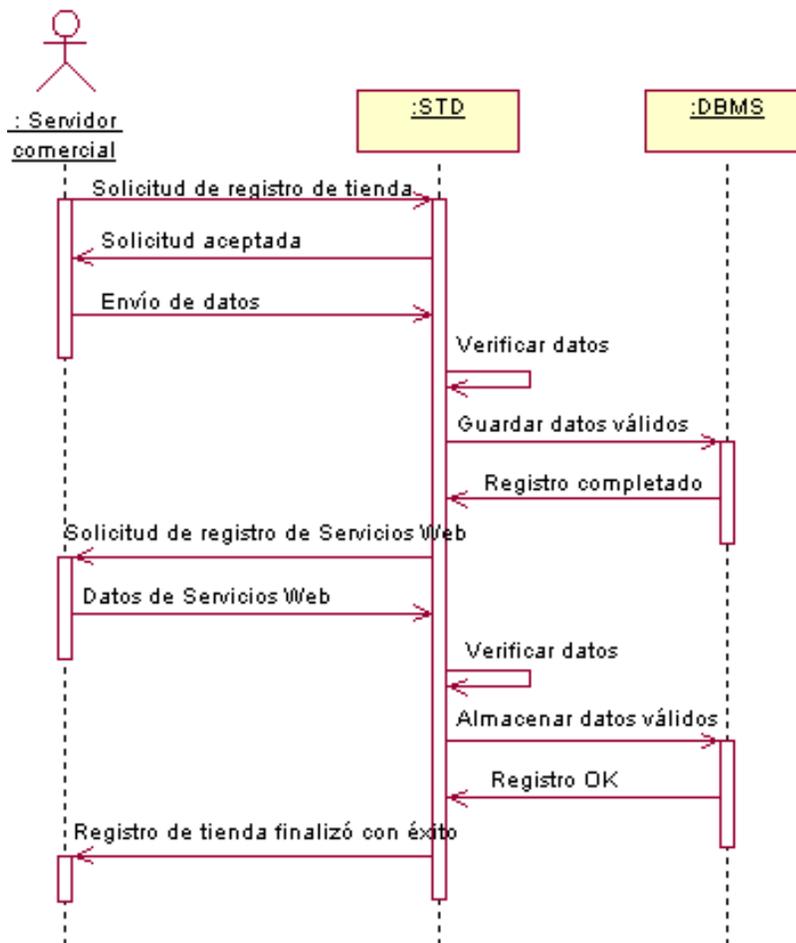


Figura 4.10: Diagrama de secuencia de registro de tiendas

categoría que eligió el cliente no cuenta con subcategorías, STD busca los productos que se encuentran en la categoría seleccionada y los muestra al cliente. Este segundo caso específicamente se observa en la figura 4.12.

Administración de pedidos. En la operación de administración de pedidos, el cliente crea su pedido. Para crear un pedido el cliente tiene que realizar varias tareas. En la figura 4.13 se representan las tareas que el cliente realiza al crear un pedido. El cliente escoge los productos de su preferencia como se mencionó anteriormente. STD toma esta lista de productos revisando que la información sobre ellos sea correcta. Si esto ocurre, STD almacena la lista en una base de datos temporal. El usuario puede ver en su navegador la lista de productos que integra su pedido.

Una vez que se completó su lista, el cliente tiene la opción de modificar alguno de los

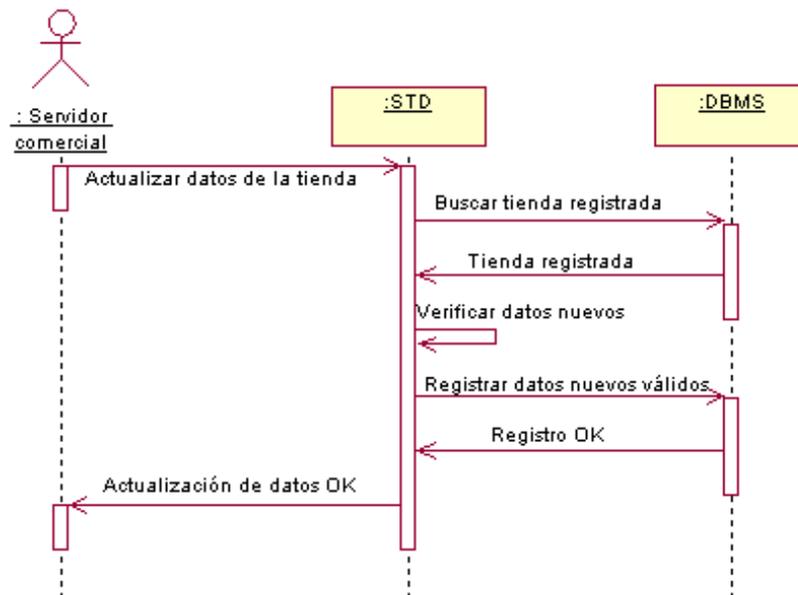


Figura 4.11: Diagrama de secuencia de actualización de datos de tienda

productos que se encuentran en ella. En caso de que el cliente desee modificar alguno de los productos de su pedido, solicitará a STD la actualización del producto. STD actualiza el nuevo registro en la base de datos temporal. El pedido está terminado en el período en que el cliente decida que la lista de productos que completa su pedido no requiere de ninguna modificación (ver figura 4.13).

Búsqueda de productos. La operación más importante y más compleja que realiza STD es la búsqueda de los productos. Para llevar a cabo la búsqueda de productos, STD realiza el procedimiento representado en la figura 4.14.

STD necesita del pedido que previamente creó el cliente en la operación de administrar pedidos. Una vez que el cliente facilite el pedido a STD, se solicita que el cliente escoja el orden de prioridad de las tiendas en las que desea llevar a cabo la búsqueda de los productos que se encuentran en su pedido. El usuario puede seleccionar de alguna o varias de las tiendas que se encuentren registradas en STD en el momento en el que se solicita la búsqueda. El cliente debe elegir por orden de preferencia las tiendas en las que desea llevar a cabo la búsqueda del pedido y enviar este orden a STD. Una vez integrados el pedido y el orden de prioridad de las tiendas STD inicia finalmente la búsqueda de los productos.

STD identifica cual de las tiendas es la de mayor prioridad para comenzar con la

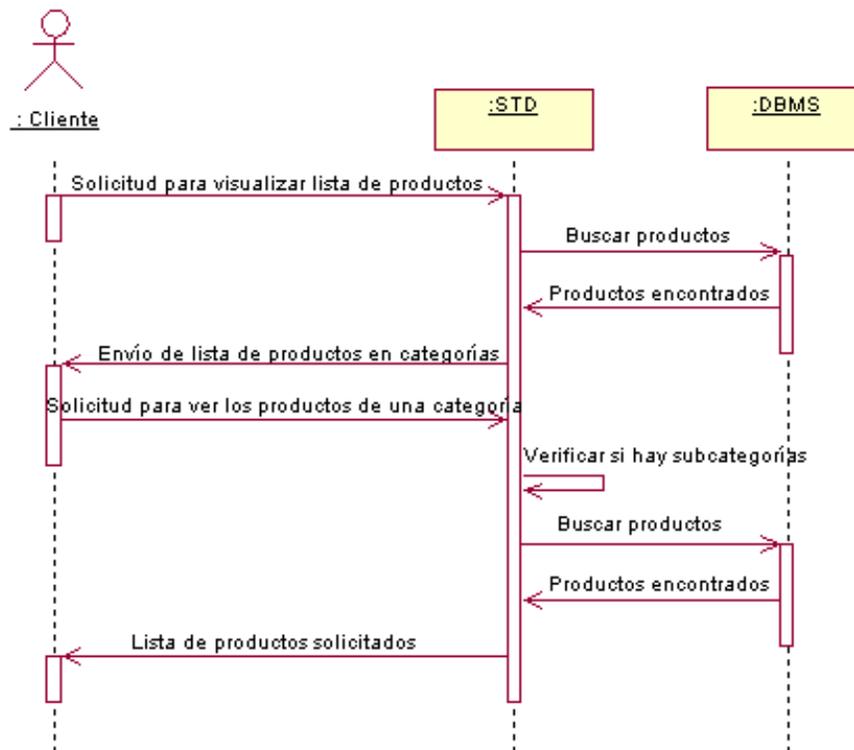


Figura 4.12: Diagrama de secuencia de administración de listas de productos

búsqueda. En la figura 4.14 la búsqueda se realiza en dos tiendas donde la de mayor prioridad será la Empresa A. STD envía el pedido a la tienda que tenga la mayor preferencia. La tienda se encargará de buscar los productos y al finalizar su búsqueda entregará a STD el resultado. STD guarda el resultado de la búsqueda en una base de datos temporal, después revisa si existe alguna otra tienda en la que se desea realizar la búsqueda, y de ser así, continúa buscando los productos en cada tienda de la misma manera finalizando con la de menor prioridad.

Próximo de tener en la base de datos el resultado de la búsqueda en cada una de las tiendas, STD envía al cliente el resultado de su búsqueda. El cliente puede visualizar el resultado en su navegador.

Visualización del resultado del pedido. Esta operación permite al cliente visualizar el resultado generado de su pedido. En la figura 4.14 podemos observar que STD envía el resultado de la búsqueda al cliente. El cliente puede ver su resultado de dos formas: en base a menor precio y en base a menor tiempo de entrega. En la figura 4.14 el cliente requiere ver su resultado en base a menor precio.

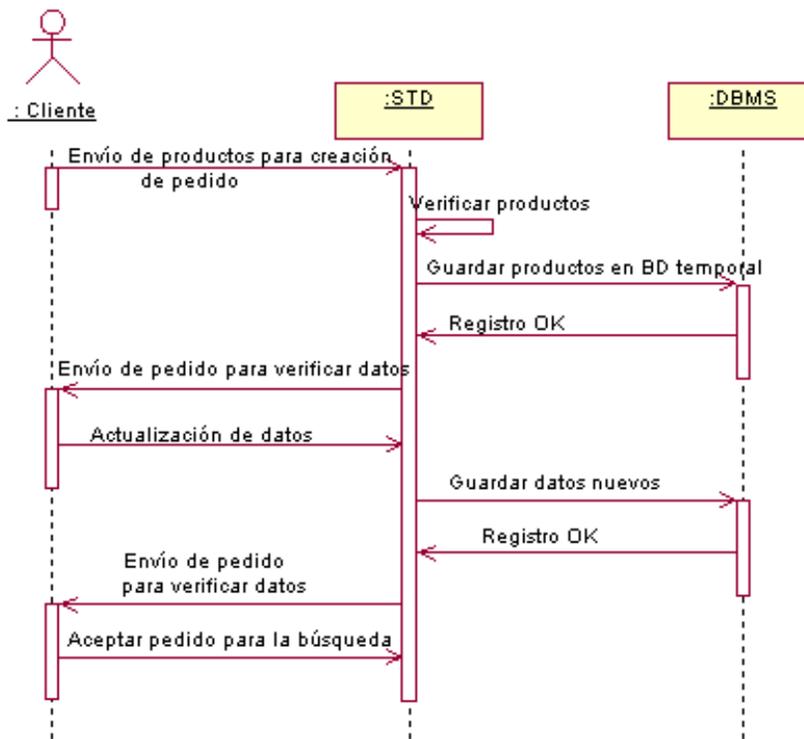


Figura 4.13: Diagrama de secuencia de administrar pedido

Cuando el cliente desea ordenar su resultado en base al menor precio, STD ordena la lista de productos que obtuvo como resultado de la búsqueda, de tal forma, en que el primer producto de la lista sea aquel que tenga el menor precio. Si se solicita ordenar en base a menor tiempo de entrega, el primer producto de la lista será el que tarde el menor tiempo en entregarse al cliente terminando la lista con el de mayor tiempo de entrega.

Administración del carrito de compra. El cliente puede agregar alguno, varios o todos los productos que se exponen en el resultado de su búsqueda a su carrito de compras. El carrito de compras representa una lista previa de los productos que el cliente quiere comprar. Las tareas que se llevan a cabo en esta operación están definidas en el diagrama 4.15. El cliente decide cuales son los productos que desea agregar a su carrito de compra. Esta nueva lista de productos se almacena en una base de datos para su posterior manipulación.

La lista de productos que contiene el carrito puede ser vista desde el navegador del cliente, de esta forma el cliente decide si desea agregar o eliminar algún producto de

su carrito. Si esto sucede, como se puede observar en la figura 4.15, STD actualiza en la base de datos el nuevo registro del producto o la eliminación de alguno ya existente. Cuando el cliente verifique su lista de productos y se encuentre plenamente convencido de que los productos de su carrito sean los que desea comprar, envía un mensaje al sistema indicándole que finalizará su compra. La compra se concluye de esta forma.

Actualización de datos. Esta operación se encuentra representada en las figuras 4.4, 4.4 y 4.13 cuando por diversas razones ya sea el cliente o la tienda desean actualizar sus datos. Las tareas que se llevan a cabo en la operación actualizar datos han sido descritas en operaciones anteriores.

Búsqueda de productos en los servidores comerciales. Esta última operación no se lleva a cabo dentro de STD sino dentro de las tiendas. La búsqueda de productos en los servidores comerciales se asemeja a la realizada por STD. La diferencia más grande es que no existe orden de prioridad para la búsqueda figura 4.16.

Para llevar a cabo la búsqueda en los servidores comerciales, estos necesitan recibir una petición de búsqueda desde STD. En esta petición STD envía el código de los productos que desea buscar. El servidor revisa mediante una consulta SQL a su base de datos la existencia del producto y, en caso de ser así, guarda la información relacionada con el mismo. El servidor envía la información de los productos solicitados a STD al final de la búsqueda.

4.5. Resumen

En este capítulo se describió el diseño del sistema STD. El sistema fue evaluado usando cuatro enfoques: organizacional, funcional, modelo de datos y modelo de comportamiento. Al inicio del capítulo, se definieron los componentes que constituyen a STD. Se puntualizaron todas las operaciones que componen al sistema y a los servidores comerciales. Se definió el diseño de las bases de datos tanto de STD como de los 3 servidores comerciales utilizados en el desarrollo del sistema. Se describió el objetivo de cada una de las tablas que componen a las diferentes bases de datos del sistema y de las tiendas. Asimismo, especificamos las relaciones entre ellas y definimos sus campos. Al final del capítulo, describimos el procedimiento que desarrollan cada una de las operaciones definidas en el enfoque funcional.

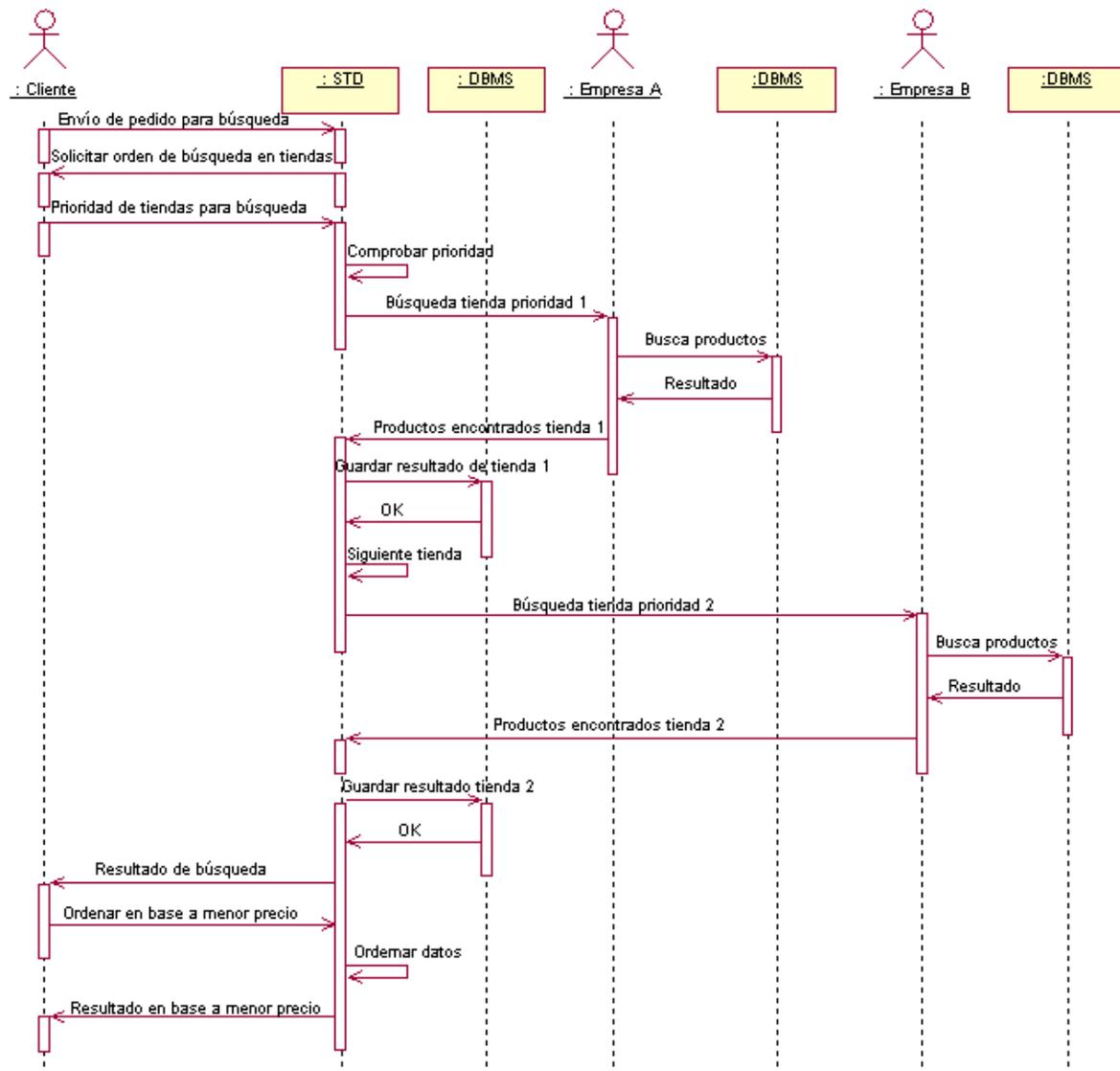


Figura 4.14: Diagrama de secuencia de búsqueda de productos

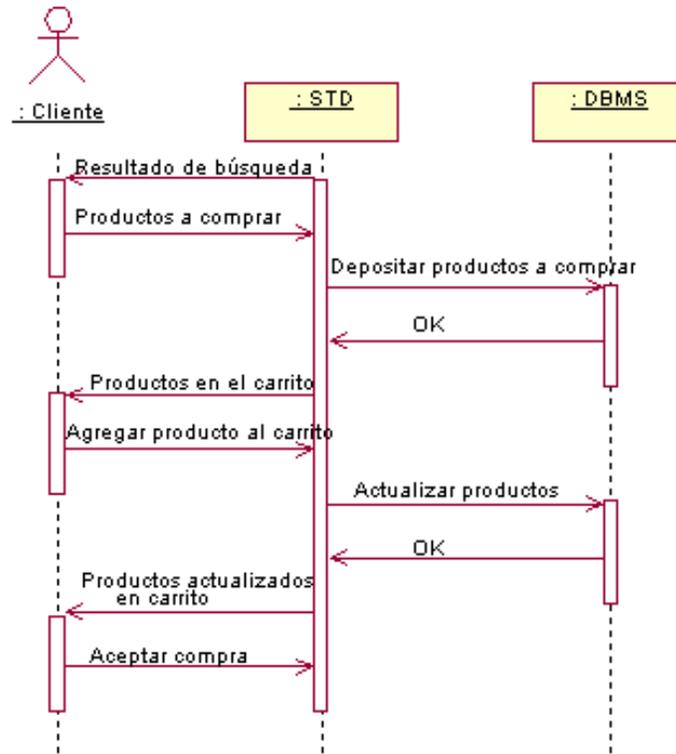


Figura 4.15: Diagrama de secuencia de administración del carrito de compras

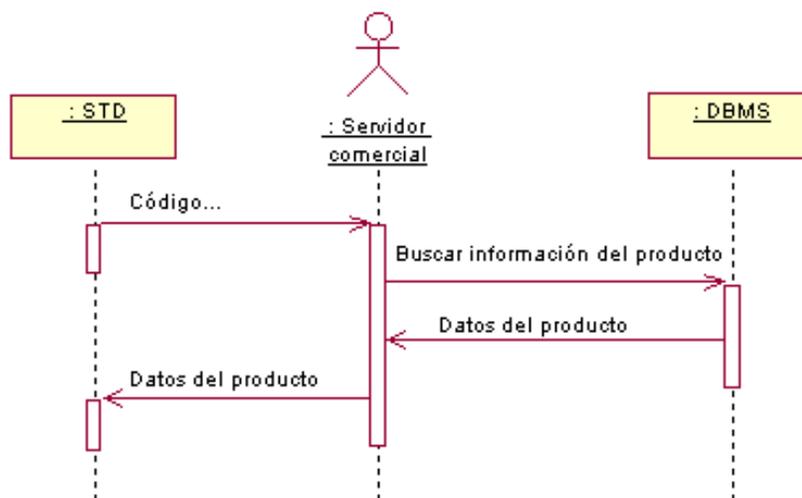


Figura 4.16: Diagrama de secuencia de búsqueda de productos en servidores comerciales

Capítulo 5

Implementación del sistema STD

En el presente capítulo se describe la implementación del sistema STD. La descripción de la implementación se hace utilizando los enfoques introducidos en el capítulo anterior organizacional, funcional, datos y comportamiento. El enfoque organizacional puntualiza las clases que constituyen a cada componente del sistema. El enfoque funcional detalla las responsabilidades que cumple cada una de las clases de los componentes de STD. Para una mejor comprensión utilizamos tarjetas CRC (Clase-Responsabilidad-Colaboración) al especificar el enfoque funcional. La implementación del enfoque del modelo de datos define detalles relativos a las bases de datos del sistema y de los servidores comerciales. Al final del capítulo se habla del enfoque del modelo de comportamiento al describir los métodos principales que implementan el funcionamiento de las clases de STD y los servidores comerciales.

5.1. Descripción organizacional

En el capítulo anterior se definieron y explicaron los 3 componentes que integran el sistema. Como se explicó, tanto STD como los servidores comerciales ofrecen diferentes servicios que conjuntamente responden a las peticiones elaboradas por los clientes. Para cumplir con su tarea, cada uno de los servicios lleva a cabo diferentes operaciones. Estas operaciones son procesadas internamente ya sea dentro de STD o dentro de los servidores de las tiendas según sea el caso, con el objeto de lograr el resultado deseado. Todas las operaciones se encuentran contenidas en los métodos de una clase. Para el funcionamiento de cada servicio se desarrollaron una o varias clases dependiendo del servicio desarrollado.

En la figura 5.1 se muestra la organización que se establece entre un cliente, STD y un servidor comercial. Las clases que forman tanto a STD como a la tienda también se muestran en la figura. Las clases se agrupan por el servicio que las integra, es decir, todas las clases que desarrollan la actividad de un servicio se encuentran representadas en un mismo rectángulo. Cada servicio se identifica por que su nombre está subrayado dentro del rectángulo. Las clases que componen el servicio se encuentran en la parte inferior del nombre del servicio (que está subrayado) definidas por su nombre. Como se puede ver en la figura 5.1, la clase que integra el servicio *Registro de clientes* que ofrece STD se llama Registro.

Además de las clases que componen los servicios, en la figura 5.1 se encuentran otros rectángulos que contienen un nombre escrito con letras mayúsculas. Estas figuras representan clases independientes que conjuntamente con los servicios responden a las peticiones de los clientes. En otras palabras, los rectángulos que contienen letras mayúsculas representan las clases que trabajan conjuntamente con las clases contenidas en los servicios para responder a las solicitudes de los clientes. Algunas de las clases que se localizan en los servicios se auxilian de las clases independientes para responder a la petición. Este tema se aborda con más detalle en las siguientes secciones.

Finalmente, se observa que el recuadro que engloba a la tienda contiene una figura en forma de elipse. La elipse simboliza al servicio Web que se localiza en la tienda y al que se invoca desde STD.

5.2. Descripción funcional

STD está integrado por un conjunto de clases Java al igual que cada uno de los servidores comerciales. Como se mencionó en la sección 4.1, STD se implementó con la tecnología JSP (Java Server Pages) teniendo como servidor Web Tomcat 5.0.9. Cada una de las páginas JSP que compone el sistema se encuentran localizadas en el servidor Tomcat, desde el cual responden a las solicitudes elaboradas por los clientes. Cabe mencionar que el proceso de compilación de las páginas JSP consiste en analizar su contenido buscando etiquetas JSP y traduciendo éstas a código Java equivalente. El contenido estático de las páginas (código html) se traduce a cadenas de caracteres en lenguaje Java. Por lo antes mencionado, cada una de las páginas se convierte en una clase Java (extensión .class) al momento de su compilación. Las clases que componen a STD y que se representan en la figura 5.1 se localizan en el servidor Tomcat.

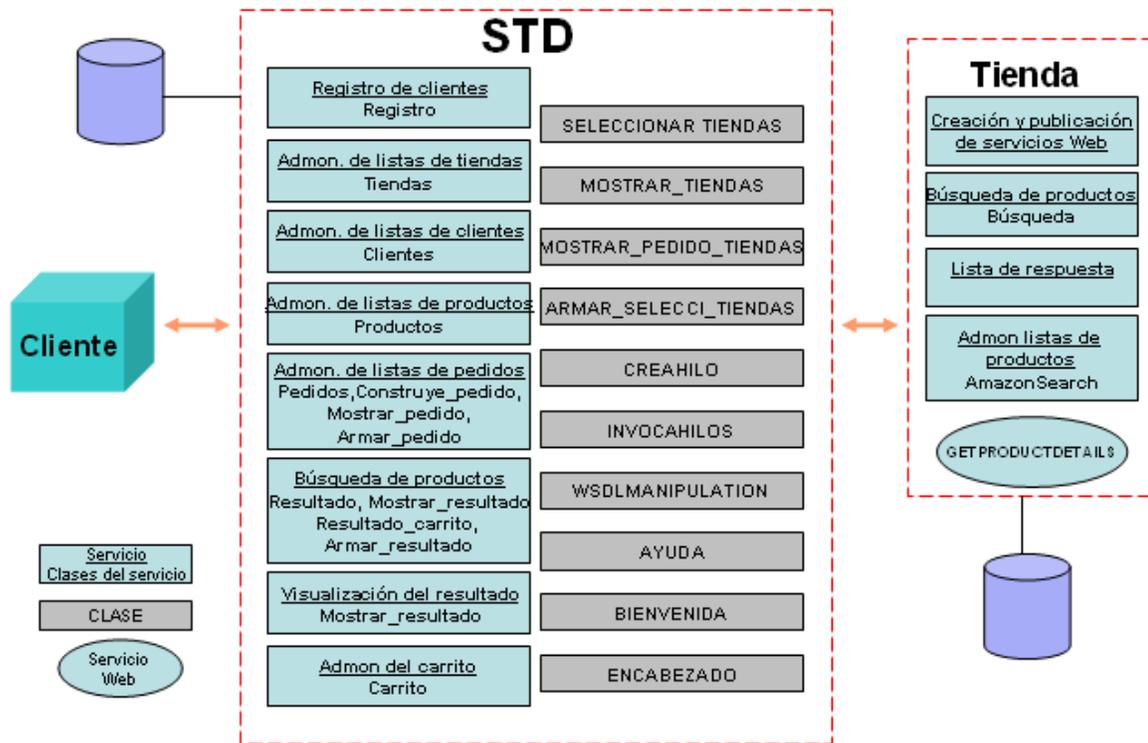


Figura 5.1: Clases de STD y de los servidores comerciales

Las clases que se crearon para el funcionamiento de los servidores comerciales fueron desarrolladas en el lenguaje de programación Java. Cada uno de los servicios Web que provee cada tienda fue creado y publicado utilizando la herramienta para desarrollo de servicios Web llamada *Web Services Developer Kit 5.0.1* de IBM. Además de ello, se utilizó la tecnología Java para desarrollar la funcionalidad principal de cada servicio.

5.2.1. Descripción de las clases de STD

Cada vez que un cliente utiliza alguno de los servicios de STD, las clases Java que se encuentran en el servidor Tomcat responden a esta solicitud. Como se explicó al inicio del capítulo, cada servicio está compuesto por diversas clases que trabajan conjuntamente para el logro de los resultados. En las secciones siguientes se explica el funcionamiento de cada una de las clases que componen los servicios de STD y de igual forma, se explican aquellas clases que trabajan ya sea en forma independiente o conjuntamente con las clases que se encuentran en los servicios.

Tabla 5.1: Clase Registro

<i>Responsabilidades</i>	<i>Colaboradores</i>
Recibir los datos del cliente desde el formulario de la página Revisar la validez de los datos aportados por el cliente Establecer la conexión con la base de datos de STD Almacenar un nuevo registro en la base de datos de STD Finalizar la conexión con la base de datos de STD Abrir la tabla de Clientes para comprobar que el cliente no esté registrado	DBMS

Registro de clientes

Este servicio se ejecuta en la clase *Registro* (ver tabla 5.1). Cuando un nuevo usuario desea ingresar al sistema debe llevar a cabo su registro, al solicitar el registro se invoca a esta clase. Su propósito es actualizar la tabla Clientes de la base de datos de STD al agregar un nuevo registro.

Administración de listas de tiendas

El servicio de administración de listas de tiendas se ejecuta en la clase *Tiendas* cuyo CRC se muestra en la tabla 5.2. Esta clase se encarga de presentar la información de las tiendas registradas en el momento en que se solicite. Además de ello, registra nuevas tiendas y actualiza su registro en caso de así solicitarlo almacenando la información necesaria según sea el caso.

Administración de listas de clientes

La clase *Clientes* (ver tabla 5.3) actualiza los registros de los clientes que así lo necesiten. El cliente solicita la modificación de sus datos, STD busca el registro y guarda la nueva información en la bases de datos de STD.

Tabla 5.2: Clase Tiendas

<i>Responsabilidades</i>	<i>Colaboradores</i>
Recibir los datos del cliente desde el formulario de la página Revisar la validez de los datos proporcionados por la tienda Establecer la conexión con la base de datos de STD Abrir la tabla de tiendas para verificar el registro de la tienda Almacenar o actualizar un registro en la base de datos de STD Enviar la información de las tiendas existentes en la base de datos al cliente Finalizar la conexión con la base de datos de STD	DBMS

Tabla 5.3: Clase Clientes

<i>Responsabilidades</i>	<i>Colaboradores</i>
Recibir la petición de actualización de información del cliente Revisar la validez de los datos proporcionados por el cliente Establecer la conexión con la base de datos de STD Abrir la tabla de clientes con el objeto de verificar la existencia del registro del cliente Actualizar el registro del cliente en la base de datos de STD Finalizar la conexión con la base de datos de STD	DBMS

Tabla 5.4: Clase Productos

<i>Responsabilidades</i>	<i>Colaboradores</i>
Recibir la solicitud de búsqueda de las subcategorías de una determinada categoría desde la página de productos Establecer la conexión con la base de datos de STD Abrir la tabla de Subcategorías para identificar si la categoría cuenta con ellas Mostrar las subcategorías de la categoría elegida Abrir la tabla de Productos y seleccionar los productos de la categoría seleccionada Mostrar los productos contenidos en la subcategoría o en la categoría Cerrar la conexión con la base de datos de STD	DBMS

Administración de listas de productos

El servicio que administra listas de productos se localiza en la clase *Productos* (tabla 5.4). La clase muestra la información de los productos solicitados según la categoría y subcategoría elegidas si fuera el caso.

Administración de pedidos

La clase *Pedidos* (ver tabla 5.5) crea y administra un pedido antes de comenzar su búsqueda. El cliente escoge los productos que contendrá el pedido. STD almacena esta lista en una tabla temporal de productos a buscar. El cliente puede modificar las cantidades y productos de su lista antes de finalizar el pedido. Para realizar estas operaciones se auxilia de las clases *Construye_pedido*, *Mostrar_pedido* y *Armar_pedido*.

La clase *Construye_pedido* que se muestra en la tabla 5.6, se encarga de tomar los datos de los productos que constituirán el pedido y de enviarlos a la clase *Pedidos* para que sean almacenados en la tabla temporal de productos a guardar. La clase *Mostrar_pedido* (ver tabla 5.7) muestra al cliente la lista de los productos que componen su pedido en caso de haber alguno. La clase *Armar_pedido* (tabla 5.8) tiene la finalidad de mandar a llamar la clase *Encabezado* e incluirla en la página de Pedidos. La clase *Encabezado* se define más adelante.

Tabla 5.5: Clase Pedidos

<i>Responsabilidades</i>	<i>Colaboradores</i>
Recibir los productos que componen el pedido desde la página de productos o desde la página de pedidos	Productos
Establecer conexión con la base de datos de STD	Construye_Pedido
Abrir la tabla temporal de la base de datos de STD	Mostrar_pedido
Almacenar los productos en la tabla	Armar_pedido

Tabla 5.6: Clase Construye_pedido

<i>Responsabilidades</i>	<i>Colaboradores</i>
Recibir los productos que el cliente solicite para agregar al pedido	Pedidos
Enviar los productos a la clase Pedidos para que sean almacenados en la base de datos	

Tabla 5.7: Clase Mostrar_pedido

<i>Responsabilidades</i>	<i>Colaboradores</i>
Establecer conexión con la base de datos de STD	Pedidos
Ejecutar consulta SQL a la tabla de Productos buscar para obtener los productos del pedido	DBMS
Mostrar los resultados de los productos que se encuentran en el pedido del cliente	
Finalizar conexión con la base de datos de STD	

Tabla 5.8: Clase Armar_pedido

<i>Responsabilidades</i>	<i>Colaboradores</i>
Llamar a la clase Encabezado e incluirla en la página de Pedidos	Pedidos

Tabla 5.9: Clase Resultado

<i>Responsabilidades</i>	<i>Colaboradores</i>
Establecer conexión con la base de datos de STD	Seleccionar_tiendas
Tomar de la tabla temporal de productos a buscar los productos que componen el pedido	InvocaHilos
Recibir desde la página de Seleccionar tiendas las tiendas en donde se buscarán los productos	CreaHilo
Ordenar las tiendas en base a su prioridad para la búsqueda	WSDLManipulation
Abrir la tabla de servicios Web y buscar las direcciones URL de las tiendas a las que invocará el servicio	Mostrar_resultado
Crear un objeto de la clase CreaHilo	Resultado_carrito
Llama al método llamadaHilo y le manda los datos de los productos y el URL del servicio a invocar y así realizar la búsqueda	Armar_resultado
Realizar las búsquedas de los productos en la tiendas invocando a sus servicios Web según su prioridad	DBMS
Almacenar los resultados de la búsqueda en la tabla temporal de precios encontrados	
Mostrar los resultados de la búsqueda obtenidos al cliente	
Finalizar la conexión con la base de datos de STD	

Búsqueda de productos

El servicio más complejo e importante con el que cuenta STD es el de búsqueda de productos; este servicio se desarrolla en la clase *Resultado* (tabla 5.9). La clase *Resultado* consulta a la tabla temporal de productos a buscar y toma los productos del pedido. Además de ello, recibe las tiendas en las buscará los productos de la clase *Seleccionar_tiendas*. En base a las prioridades de búsqueda proporcionadas por el cliente, busca en las diferentes tiendas el pedido elaborado por el cliente y almacena los resultados en la tabla temporal de precios encontrados auxiliándose de las clases *InvocaHilos*, *CreaHilo* y *WSDLManipulation*.

La clase *Resultado* manda llamar a las clases *Mostrar_resultado*, *Resultado_carrito* y *Armar_resultado*. La clase *Mostrar_resultado* (ver tabla 5.10) presenta al cliente los resultados de la búsqueda de su pedido cuando esta ha finalizado y ordena el resultado

Tabla 5.10: Clase *Mostrar_resultado*

<i>Responsabilidades</i>	<i>Colaboradores</i>
Establecer conexión con la base de datos de STD Consultar a la tabla de tiendas y guardar sus datos en un arreglo Verificar si se seleccionó alguno de los criterios de búsqueda Consultar a la base de datos de Precios encontrados con el objeto de obtener los resultados de la búsqueda Mostrar los resultados de la búsqueda al cliente ya sea ordenados por el criterio seleccionado ó por el predefinido por el sistema Finalizar la conexión con la base de datos de STD	Resultado DBMS

en base a menor precio ó menor tiempo de entrega. *Resultado_carrito* (tabla 5.11) indica al cliente la totalidad de los productos que ha elegido para que sean guardados en su carrito de compra. Además de ello, checa que la cantidad pedida por el cliente satisfaga la cantidad en existencia en la tienda; de no ser así envía al cliente un mensaje. La clase *Armado_resultado* que se observa en la tabla 5.12 fue creada con el objeto de incluir la clase *Encabezado* en la página de Resultados.

La clase *Seleccionar_tiemdas* (ver tabla 5.13) muestra al cliente un listado de los productos que contiene su pedido y, además, el cliente puede ver la lista de tiendas en la que puede realizar su búsqueda. La clase *Seleccionar_tiemdas* envía las prioridades de búsqueda de las tiendas a la clase *Resultado*. *Seleccionar_tiemdas* trabaja conjuntamente con las clases *Mostrar_tiemdas*, *Mostrar_pedido_tiemdas* y *Armado_seleccionar_tiemdas*.

La clase *Mostrar_tiemdas* (ver tabla 5.14) se encarga de mostrar el listado de las opciones de tienda en las que el cliente puede efectuar la búsqueda de productos.

La clase *Mostrar_pedido_tiemdas* proporciona al cliente una lista de los productos que contiene su pedido, si es que ha registrado alguno. Las responsabilidades de esta clase son iguales a las que ofrece la clase *Mostrar_pedido*, lo único que varía en ambas es la presentación que el cliente visualiza de ellas en su navegador. La clase *Armado_seleccionar_tiemdas* que se observa en la tabla 5.15 al igual que algunas de las clases mencionadas anteriormente, incluye la clase *Encabezado* dentro de la página de *Seleccionar_tiemdas*.

Tabla 5.11: Clase Resultado_carrito

<i>Responsabilidades</i>	<i>Colaboradores</i>
Establecer conexión con la base de datos de STD Consultar a la tabla de Tiendas y guardar los datos de las tiendas en un arreglo Consultar a la tabla de Carrito para verificar los productos que fueron enviados al carrito de compra Revisar si la cantidad existente en la tienda no superó la cantidad solicitada por el cliente Mostrar los resultados de los productos existentes en el carrito Finalizar conexión con la base de datos de STD	Resultado DBMS

Tabla 5.12: Clase Armar_resultado

<i>Responsabilidades</i>	<i>Colaboradores</i>
Llamar a la clase Encabezado e incluirla en la página de Resultado	Resultado

Tabla 5.13: Clase Seleccionar_tiendas

<i>Responsabilidades</i>	<i>Colaboradores</i>
Establecer conexión con la base de datos de STD	Resultado
Verificar si el cliente seleccionó alguno de los productos del pedido para eliminar	Mostrar_tiendas
Eliminar de la tabla de Productos buscar el producto	Mostrar_pedido_tienda
Mostrar al cliente un listado de los productos que contiene el pedido	Armar_seleccionar_tiendas
Mostrar la lista de tiendas en la que se puede realizar la búsqueda	DBMS
Finalizar la conexión con la base de datos de STD	

Tabla 5.14: Clase Mostrar_tiendas

<i>Responsabilidades</i>	<i>Colaboradores</i>
Establecer conexión con la base de datos de STD Realizar una consulta SQL para obtener las tiendas registradas en STD desde la tabla de tiendas Mostrar las tiendas con el orden de prioridad predeterminado por el sistema al cliente Finalizar la conexión con la base de datos de STD	Seleccionar_tiendas DBMS

Tabla 5.15: Clase Armar_seleccionar_tiendas

<i>Responsabilidades</i>	<i>Colaboradores</i>
Llamar a la clase Encabezado e incluirla en la página de Seleccionar tiendas	Seleccionar_tiendas

La clase *CreaHilo* representada en la tabla 5.16 crea un hilo para cada una de las búsquedas. Se crea un hilo para cada tienda. Cuando las tiendas son de igual prioridad se crea un hilo para la totalidad de las tiendas con dicha prioridad, es decir, si dos tiendas tienen la misma prioridad se crea un solo un hilo en donde realizarán la búsqueda las 2 tiendas. La ejecución del hilo termina cuando finalice la búsqueda del pedido en la tiendas. En el caso de tiendas con igual prioridad, los hilos finalizarán hasta que la búsqueda en todas ellas termine. El hilo se construye a partir de los parámetros enviados por la clase *Resultado*. Estos parámetros incluyen la lista de productos (el pedido) a buscar, la(s) tienda(s) en que se realizará la búsqueda y la(s) dirección(es) del(los) servicio(s) Web donde se hará la invocación.

La clase *InvocaHilos* mostrada en la tabla 5.17 ejecuta el hilo creado en la clase *CreaHilo*. Esta clase toma del hilo los parámetros guardados en la clase *CreaHilo* y los extrae. En base a la información almacenada en el hilo comienza la búsqueda en la tienda y dirección dadas. La clase invoca al servicio Web y le envía la solicitud de búsqueda de productos. Al finalizar la búsqueda, los resultados se almacenan en la tabla temporal de Precios encontrados.

La clase *WSDLManipulation* (ver tabla 5.18) convierte un mensaje SOAP a una cadena. El método que realiza esta función se llama desde la clase *InvocaHilos*.

Tabla 5.16: Clase CreaHilo

<i>Responsabilidades</i>	<i>Colaboradores</i>
<p>Recibir los parámetros de productos a buscar, tiendas y URL donde se invocará el servicio desde la clase Resultados</p> <p>Crear un hilo para cada una de las tiendas, el nuevo hilo tendrá los valores del identificador de la tienda, el URL y los productos</p> <p>Ejecutar el hilo</p> <p>Al finalizar el hilo mandar un mensaje de terminación</p>	<p>Resultado</p> <p>InvocaHilos</p>

Tabla 5.17: Clase InvocaHilos

<i>Responsabilidades</i>	<i>Colaboradores</i>
<p>Extraer los parámetros con la información del hilo</p> <p>Invocar al servicio Web enviándole la solicitud de búsqueda de productos</p> <p>Establecer conexión con la base de datos de STD</p> <p>Verificar que el producto encontrado no exista en la tabla de precios encontrados</p> <p>Almacenar en la tabla de precios encontrados los resultados de la búsqueda</p> <p>Finalizar la conexión con la base de datos de STD</p>	<p>WSDLManipulation</p> <p>DBMS</p>

Tabla 5.18: Clase WSDLManipulation

<i>Responsabilidades</i>	<i>Colaboradores</i>
<p>Recibir un mensaje de tipo SOAP</p> <p>Convertir el mensaje SOAP a una cadena</p> <p>Regresar la cadena a la clase InvocaHilos</p>	<p>InvocaHilos</p>

Tabla 5.19: Clase Carrito

<i>Responsabilidades</i>	<i>Colaboradores</i>
Establecer conexión con la base de datos de STD Abrir la tabla temporal de carrito Seleccionar todos los productos de la tabla Realizar la suma de la compra Mostrar al cliente los productos del carrito conjuntamente con el total de la compra Cerrar la conexión con la base de datos de STD	DBMS

Visualización del resultado del pedido

El servicio de visualización del resultado pedido se desarrolla en la clase *Mostrar_resultado*. Esta clase se encarga de mostrar al cliente los resultados de la búsqueda que solicitó. Hablamos de esta clase en la sección correspondiente a *Búsqueda de productos*.

Actualización de datos

El servicio de actualización de datos se realiza en las clases *Clientes*, *Pedidos* y *Tiendas* descritas en las secciones anteriores. Las clases reciben desde el navegador del cliente la petición de actualización de datos. Se revisa que los datos existan en las bases de datos. Los nuevos datos se almacenan en las tablas correspondientes.

Administración del carrito de compras

El servicio de administración del carrito de compras se utiliza en la clase *Carrito* (ver tabla 5.19). La clase hace una consulta a la base de datos y muestra al cliente todos los productos que agregó a su carrito de compra. Asimismo, indica el total a pagar por la compra antes de finalizarla.

5.2.2. Descripción de las clases de los servidores comerciales

Como mencionamos en el capítulo 4 los servidores comerciales ofrecen 4 servicios: creación y publicación de servicios Web, búsqueda de productos solicitados, formar lista de respuesta a la búsqueda solicitada y administración de listas de productos. Para poder crear y publicar un servicio Web no se desarrolló clase Java alguna, para ello utilizamos

el servidor de IBM mencionado previamente. A continuación se describen las clases que se encuentran en los servicios que ofrecen los servidores comerciales. De igual forma, se describe el servicio Web que fue desarrollado para la búsqueda de productos.

Búsqueda de productos solicitados

Cuando finaliza la creación de un servicio Web utilizando el servidor de IBM, se crea una carpeta que tiene el nombre elegido por el desarrollador. Esta carpeta, contiene toda la información correspondiente al servicio Web desarrollado. Dentro de esta carpeta se encuentra otra carpeta que lleva por nombre *classes*. La carpeta contiene un archivo Java, el archivo se puede identificar porque lleva al final de su nombre la palabra *BindingImpl*. Este archivo representa la principal funcionalidad del servicio Web creado previamente. El archivo tal y como fue creado por el servidor no tiene funcionamiento alguno. Para brindarle funcionamiento debe modificarse. El desarrollador modifica el archivo con el propósito de que realice lo que se espera del servicio Web al invocarlo.

Este archivo Java fue modificado de diferente forma en cada uno de los servicios Web realizados, esto debido a que cada tienda contiene diferentes datos. Por esta razón no se pudo generalizar la clase. La clase Java fue modificada con el objetivo de que proporcionara el servicio de búsqueda de productos. La búsqueda de productos que realiza cada servicio Web es esencialmente igual para todos, la diferencia principal radica en el nombre de las variables que dependen directamente de las nombradas en las bases de datos de las tiendas.

A continuación se enumeran los pasos desarrollados por la clase que implementa la principal funcionalidad de los servicios Web de los servidores comerciales desarrollados para probar la funcionalidad de STD:

1. Se crea un objeto de la clase que contiene los métodos: nombre producto, cantidad, precio, y tiempo de entrega. Los métodos son nombrados de forma diferente en cada tienda.
2. Se realiza la conexión a la base de datos de la tienda.
3. Se hace una consulta a la tabla que contiene los productos de la tienda.
4. La consulta solicita los siguientes datos: nombre del producto, cantidad, precio, tiempo de entrega del producto a buscar.
5. Los resultados se guardan en los métodos del objeto creado.

6. La clase regresa los resultados encontrados en la búsqueda.

Todas las clases Java modificadas para brindar el servicio de búsqueda de productos en los servidores comerciales tienen el nombre de *GetProductDetailsBindingImpl*. Aunque el nombre es el mismo para todas ellas el contenido varía ligeramente.

Formar lista de respuesta solicitada

Este servicio se cumple cuando una vez finalizada la búsqueda de productos en los servidores comerciales, el resultado del servicio es encapsulado en un mensaje SOAP y enviado directamente a STD. STD lo convierte a cadena y analiza el resultado. Para este servicio no se creó clase Java alguna.

Administrar listas de productos

En el capítulo 4 mencionamos que las empresas A y B poseen una base de datos con estructura similar o idéntica y que esta base de datos se asemeja al diseño de la base de datos de Amazon. Para diseñar estas bases de datos se analizó la información suministrada por los servicios Web de Amazon. Para invocar a estos servicios Web y obtener de ellos información real de Amazon se utilizó y modificó la clase *AmazonSearch* (ver tabla 5.20).

La clase *AmazonSearch* invoca al servicio Web de *Amazon* que se encuentra especificado en "http://soap.amazon.com/schemas2/AmazonWebServices.wsdl". Amazon realiza diferentes tipos de búsqueda. Este servicio Web realiza la búsqueda de productos por palabra clave. Esta búsqueda se encarga de encontrar en las bases de datos de Amazon, los productos que se asemejen a la palabra dada por el cliente. Los parámetros que recibe el servicio Web son la palabra clave, el catálogo y un número. La palabra clave corresponde al producto que busca el servicio de Amazon. El catálogo puede definirse como un tipo determinado de categoría y el número indica la cantidad de productos que el cliente desea ver. El número uno muestra 10 resultados, el dos muestra 20 resultados y así sucesivamente.

Utilizando esta clase se analizaron los resultados arrojados por Amazon. Como se mencionó en el capítulo 4 las bases de datos de estos servidores contienen 14 tablas. Doce de ellas, fueron diseñadas en base a los resultados obtenidos de Amazon. La clase *AmazonSearch* se modificó con la finalidad de que una vez terminado el diseño de las bases de datos de los servidores, los datos obtenidos del servicio Web de Amazon fueran almacenados en las bases de datos de los servidores. Logrando con ello, que las bases de

Tabla 5.20: Clase AmazonSearch

<i>Responsabilidades</i>	<i>Colaboradores</i>
Crea un objeto de la clase GuardaDatos	GuardaDatos
Crea un objeto de la clase AmazonSearch	Stock
Utiliza un método de la clase AmazonSearch y le envía los datos de la búsqueda	Track
Invoca al servicio Web de Amazon	Details
Recibe los resultados de Amazon	Reviews
Guarda en las bases de datos de los servidores A y B la información encontrada de los productos	ThirdPartyProductInfo
Muestra los resultados encontrados	BrowseNode
	CustomerReview
	KeyPhrase
	ThridPartyProductDetails

datos contengan datos reales. Todos los datos que contienen las bases de datos de las empresas A y B son datos reales que se pueden encontrar en Amazon. El servicio que administra las listas de productos de los servidores comerciales de las empresas A y B está implementado por la clase *AmazonSearch*.

Para el servidor comercial *Gigante STD* no se desarrolló la clase que administra listas de productos. Como se mencionó en la sección 4.3.2 estas tablas fueron diseñadas en base al esquema que presentan algunas organizaciones que ofrecen servicios en línea. Los productos que contiene esta tabla fueron tomados de algunas de estas organizaciones. Su registro en la base de datos fue llenado en forma manual y su administración que incluye la modificación y actualización de productos también lo es.

El Servicio Web

En las secciones anteriores se mencionó que se implementó un servicio Web para cada uno de los servidores comerciales. Este servicio Web realiza la búsqueda de productos en la tienda que lo contiene. Los 3 diferentes servicios Web creados recibieron el nombre de *GetProductDetails*, como se mencionó anteriormente aunque el nombre del servicio es el mismo, es diferente para cada una de las tiendas. Cabe mencionar que el servidor de IBM se instaló en la máquina donde se desarrolló el proyecto de tesis. En ese mismo servidor se crearon y publicaron los servicios. Los servicios se pueden invocar desde las siguientes direcciones:

Empresa A

"http://192.168.50.13:6080/EmpresaA_1/services/getProductDetails"

Empresa B

"http://192.168.50.13:6080/EmpresaB_1/services/getProductDetails"

Gigante STD

"http://192.168.50.13:6080/GiganteSTD_1/services/getProductDetails"

En esencia el servicio Web desarrolla los pasos descritos en la sección *Búsqueda de productos solicitados*. Para lograr la búsqueda, el servicio recibe como parámetro de entrada el código de producto. Busca en las bases de datos de la tienda (donde se está llamando el servicio) los datos de ese producto y al finalizar envía los datos referentes a: nombre del producto, cantidad, precio y tiempo de entrega del producto que se buscó a quien lo invocó (en este caso STD). Estos son los datos que STD recibe y que manipula con el fin de enviarlos como respuesta a la solicitud del cliente. Para ejemplificar lo antes mencionado, en la figura 5.2 se muestra el mensaje SOAP que realiza la invocación del servicio Web de uno de los servidores comerciales. Para este ejemplo el servidor comercial es Empresa A. Cabe mencionar en este punto, que en este trabajo de investigación SOAP se usó con RPCs (para mayor información revisar sección 2.9).

```
<SOAP-ENV: Envelope SOAPENV:encodingStyle=
"http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV =
"http://schemas.xmlsoap.org/soap/envelope/">
  <SOAPENV:Body>
    <getProductDetails>
      <productCode>431718</productCode>
    </getProductDetails>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figura 5.2: Mensaje de invocación al servicio Web *getProductDetails* desde servidor comercial *Empresa A*

La figura 5.2 muestra el mensaje SOAP enviado a la Empresa A en solicitud del producto. Como se puede observar en la figura, el mensaje SOAP contiene el parámetro de entrada que necesita el servicio Web para desarrollar la búsqueda. Este parámetro consiste en el código de producto. En la figura 5.2 el código de producto está definido por el número 431718.

```

===== Inicio de respuesta =====
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas
  <soapenv:Body>
    <getProductDetailsResponse xmlns="">
      <ProductDetails>
        <productName>Computers</productName>
        <quantity>962</quantity>
        <price>60</price>
        <deliveryTime>1 semana</deliveryTime>
      </ProductDetails>
    </getProductDetailsResponse>
  </soapenv:Body>
</soapenv:Envelope>
===== Fin de respuesta =====

Process Exit...

```

Figura 5.3: Mensaje de respuesta generado desde el servidor comercial *Empresa A* al invocar su servicio Web *getProductDetails*

Una vez que la Empresa A recibe la solicitud de búsqueda, desarrolla la búsqueda correspondiente y envía el resultado de la misma a STD. En la figura 5.3 se muestra la respuesta generada por la Empresa A al finalizar la búsqueda del producto 431718. Los resultados generados indican que el producto 431718 lleva por nombre *Computers*, el producto tiene una cantidad en existencia en la tienda de 962 unidades, el precio registrado del producto es de 60. Finalmente, la respuesta indica que el tiempo de entrega del producto es de 1 semana.

5.3. Descripción de los datos

La mayor parte de los servicios que ofrecen tanto STD como de los servidores comerciales, toman y manipulan datos que obtienen de diferentes bases de datos. Las bases de datos de STD y de los servidores comerciales fueron administradas utilizando el manejador de base de datos de Microsoft Access.

Para lograr el acceso a estos datos desde JDBC (Java DataBase Connectivity) se utilizó la ODBC (Open DataBase Conectivity). JDBC es un API que permite conectarse al sistema administrador de la base de datos usando el lenguaje de consulta SQL. La ODBC es la interfaz propuesta por Microsoft para acceder a datos en un entorno de sistemas de administración de bases de datos no relacionales (DBMS). La ODBC permite que las aplicaciones accedan a múltiples fuentes de datos sin tener que comprender el formato esencial de los datos que se están leyendo.

El puente JDBC-ODBC permite a los programas Java usar manejadores ODBC para invocar fuentes de datos ODBC. Es mediante este mecanismo que tanto los servidores comerciales como STD puede acceder a sus datos. La descripción de las tablas contenidas en STD y los servidores comerciales, así como la definición de sus campos se especificó en el capítulo anterior.

5.4. Descripción del comportamiento

En la sección 5.2 hablamos sobre las clases que componen a cada uno de los servicios de STD y de los servidores comerciales. La mayoría de las clases fueron desarrolladas usando la tecnología JSP. Las páginas JSP contienen el código Java que da funcionalidad a STD. Los fragmentos de programación se encuentran incrustados en las páginas JSP. En las páginas solo incrustamos código Java cuando así se necesita. La mayor parte de las páginas JSP que contiene STD fueron realizadas sin necesidad de llamar algún método, sino que, cada vez que se necesitaba dar funcionalidad a cierta parte de la página se desarrolló el código Java capaz de lograr esta tarea.

Por lo antes mencionado, la mayor parte del funcionamiento de las clases que componen el sistema STD se encuentra dentro de las páginas JSP. El servicio más complejo que proporciona STD se implementa por la clase *Resultado*. Esta clase es la única que se auxilia de otras clases desarrolladas en lenguaje Java para cumplir con su tarea que es la de buscar productos. Las clases a las que llama Resultado son: *CreaHilo*, *InvocaHilos* y *WSDLManipulation*.

Las clases *CreaHilo*, *InvocaHilos* y *WSDLManipulation* contienen varios métodos que son los que finalmente desarrollan la principal funcionalidad del sistema que es la búsqueda de productos. Estos métodos conjuntamente con los métodos de las clases que se localizan en los servidores comerciales, se describen a continuación definidos en la clase que los contiene.

Métodos de la clase CreaHilo

La clase *CreaHilo* contiene los siguientes métodos:

- **llamadaHilo(String tienda, String direccionSW, String[] productos_solicitados).**
Este método se encarga de construir un hilo para desarrollar cada una de las búsquedas del pedido en las tiendas. El método recibe los parámetros *tienda* de

tipo cadena, *direccionSW* de tipo cadena y *productos_solicitados* de tipo arreglo de cadenas. El parámetro *tienda* corresponde al número identificador de la tienda en la que se realiza la búsqueda. Este identificador debe coincidir con alguno de los registrados en la tabla de Tiendas de la base de datos de STD. El parámetro *direccionSW* contiene la dirección URL a la que se invocará el servicio Web y finalmente el parámetro *productos_solicitados* guarda los códigos de los productos que se buscarán en la tienda indicada por el parámetro *tienda*.

El método construye un objeto de la clase *InvocaHilos*. Cabe mencionar que la clase *InvocaHilos* hereda de la clase *Thread* de Java por lo que hereda todos sus métodos. Una vez que el objeto existe, llama al método *Hilo()* de la clase con el mismo nombre y guarda en el los datos tomados por los parámetros de entrada. El método *llamadaHilo()* se encarga de ejecutar el hilo que creó llamando al método *start()*. El método *start()* es quien se encarga de poner el hilo creado por el método en ejecución. Finalmente, el método *llamadaHilo()* verifica que el hilo termine su ejecución utilizando el método *isAlive()*. Cuando el hilo ha finalizado su ejecución el método finaliza.

- **llamadaHilo(String[] [] tiendas, String[] productos_solicitados)**. La clase *CreaHilo* contiene otro método llamado de igual forma que el antes mencionado. La diferencia entre ambos radica en el tipo de parámetros que reciben. El método anterior se llama en el caso de que solo se envié para la búsqueda de una tienda. Dicho de otra forma, cuando la prioridad de las tiendas es única (no hay dos con la misma prioridad) se llamará al método descrito anteriormente. Cuando la prioridad de las tiendas sea la misma para al menos dos de ellas se llamará a este método.

Este método recibe los mismos parámetros que el anterior pero organizados de diferente manera. Debido a que este método recibe tiendas con el mismo número de prioridad, recibe el arreglo de cadenas *tiendas* en donde el primer campo contiene el número identificador de la tienda y el segundo la dirección URL donde se encuentra especificado el servicio Web. El segundo parámetro *productos_solicitados* de tipo arreglo de cadenas, contiene al igual que en el método anterior el arreglo de los códigos de los productos que se buscarán.

Este método trabaja de igual forma que el anterior, solo que en vez de crear un objeto de la clase *InvocaHilos* crea el número de objetos dependiendo de las

tiendas que contengan la misma prioridad. Si el arreglo de cadenas *tiendas* contiene 3 elementos entonces crea 3 objetos, si son 4 crea 4 objetos y así sucesivamente. Después de crear cada hilo se llama al método *Hilo()* de la clase *InvocaHilos* y se guarda en cada uno de ellos la información que corresponde al número de tienda y la dirección del servicio Web y el arreglo de productos de igual forma que en el método anterior. La forma en que se ejecuta cada hilo es igual a la mencionada en el método antes mencionado. Este método finaliza hasta que la búsqueda en cada uno de los hilos termine su ejecución. Para verificar que ninguno de los hilos exista utiliza de igual forma el método *isAlive()*.

Métodos de la clase *InvocaHilos*

La clase *InvocaHilos* contiene básicamente 2 métodos:

- **Hilo(String tienda, String direccionSW, String[] productos_solicitados).** Este método simplemente está dedicado a guardar los parámetros correspondientes a la tienda (*tienda*), dirección del servicio Web (*direccionSW*) y el arreglo de productos (*productos_solicitados*). Esto con el objeto de que puedan ser utilizados en el método *run()*, a la hora de poner en ejecución al hilo.
- **run().** El método *run()* se encarga hacer la invocación al servicio Web de la tienda y de guardar los resultados obtenidos de la búsqueda en la base de datos de STD. Este método es donde finalmente se realiza la búsqueda de los productos y se recolectan los resultados. El método *run()* trabaja básicamente de la forma que se describe a continuación. Utiliza el arreglo de *productos_solicitados* que contiene el hilo que se está ejecutando para desarrollar sus operaciones. Como se mencionó anteriormente, el arreglo de productos contiene todos los productos que componen el pedido. *Run()* toma uno a uno los productos contenidos en el arreglo y los utiliza para completar el parámetro de entrada que es enviado al servicio Web de la tienda donde se realiza la invocación. Una vez que se tengan los datos necesarios para componer el mensaje SOAP que invocará al servicios Web, se envía una solicitud a la tienda para que esta realice la búsqueda del producto que se le solicitó.

Cuando la tienda finaliza la búsqueda del producto, el método recibe la respuesta en forma de mensaje SOAP y la convierte a una cadena, auxiliándose del método *SOAPToString* que se encuentra en la clase *WSDLManipulation*. Para obtener

los resultados generados en la respuesta, de la cadena se separan las etiquetas que contienen el precio, la cantidad y el tiempo de entrega del producto. Los datos obtenidos en estas etiquetas reflejan los resultados generados por las tiendas, mismos que serán almacenados en las bases de datos para que puedan mostrarse al cliente.

Los resultados se almacenan en tablas temporales que se encuentran localizadas en la base de datos de STD. Los resultados obtenidos de todos los productos del pedido solicitado, se almacenarán en la base de datos aún cuando alguno de ellos no haya sido localizado en alguna, varias o todas las tiendas en las que se buscó. Los datos que se registran en las tablas temporales respecto de cada producto buscado son: identificador de la tienda, código de producto, precio, cantidad, tiempo de entrega y número.

El identificador de la tienda hace referencia a la tienda que generó la respuesta de la solicitud de producto. Es decir, si la tienda a la que se solicitó el pedido y que generó la respuesta es la identificada como 1 en el sistema, será el número 1 el que se guardará en este campo. El código de producto es el número de identificador con que se localiza el producto en STD. El precio, la cantidad y el tiempo de entrega son los datos que corresponden al producto buscado. Finalmente, se guarda en las bases de datos temporales un número que puede ser 0 ó 1, que indica si el producto se haya localizado en la tienda o si no lo está (para mayor información revisar sección 4.3.1).

Método de la clase WSDLManipulation

Esta clase cuenta básicamente con un método que se define en seguida:

- **SOAPToString.** Este método convierte un mensaje SOAP en una cadena. Esto se realizó con el objeto de poder manipular de una manera más sencilla la información que envían los servidores comerciales como respuesta a la búsqueda de productos.

Método de la clase AmazonSearch

Como ya hemos mencionado, la clase *AmazonSearch* es la que se encarga de invocar a uno de los servicios Web de Amazon y de obtener datos propios de dicha empresa mediante el. También mencionamos que para guardar esta información dentro de las

bases de datos diseñadas para los servidores comerciales A y B, se modificó esta clase agregando una clase llamada *GuardaDatos* quien se encarga de almacenar la información obtenida por la clase *AmazonSearch* dentro de las bases de datos de los servidores comerciales A y B. La clase *AmazonSearch* contiene varios métodos, uno de esos métodos llama a la clase *GuardaDatos*. A continuación describiremos el método más importante de la clase *AmazonSearch* y al finalizar definiremos el método de la clase *GuardaDatos*.

- **KeywordSearch(String name, String mode, String page).** Este método se encarga de invocar al servicio Web de Amazon que ofrece la búsqueda mediante palabra clave. Los parámetros que recibe este método son *name* que hace referencia al nombre del producto ó palabra clave que desea buscar. El parámetro *mode* es un tipo de catálogo definido por Amazon, podríamos asemejarlo a un tipo de categoría. Finalmente, *page* hace referencia a la totalidad de resultados que se quieren obtener. Ejemplificando esto, si queremos obtener 10 resultados *page* tendrá el valor de 1, si deseamos obtener 20 resultados tendrá el valor de 2.

Método de la clase **GuardaDatos**

Esta clase contiene un único método que es el siguiente:

- **guardar (Details[] detalles).** Este método se encarga de almacenar en las bases de datos de los servidores toda la información que se obtuvo de la invocación del servicio Web de Amazon. El parámetro que recibe este método *detalles*, contiene toda la información que se obtuvo en la respuesta. Es de este arreglo de donde se toman todos los datos que se almacenarán en las diferentes tablas de las bases de datos de los servidores comerciales. Primeramente, se establece la conexión con la base de datos del servidor comercial donde se guardarán los datos. El método extrae cada uno de los elementos del arreglo *detalles* y los va almacenando en las bases de datos. Al finalizar se cierra la conexión con la base de datos.

5.5. Resumen

En este capítulo detallamos la implementación del sistema STD cuyo diseño se explicó en el capítulo 4. Describimos las clases que componen los servicios proporcionados por los servidores comerciales y STD. Para un mejor entendimiento de las clases, se utilizaron tarjetas CRC. Se describió la implementación del almacenamiento de los datos en las bases de datos. Al finalizar el capítulo se detallaron los métodos desarrollados en las clases Java que soportan los servicios de STD y de los servidores comerciales.

Antes de concluir, es necesario mencionar en este punto, que el código Java de las clases *AmazonSearch* y *WSDLManipulation* fue desarrollado por Giner Alor Hernández. El código de la clase *AmazonSearch* se modificó con la finalidad de adaptarlo a las necesidades que demandó el desarrollo del sistema. De la clase *WSDLManipulation* solo se utilizó para fines del sistema, el código correspondiente al método *SOAPtoString*.

Capítulo 6

Conclusiones y trabajo a futuro

6.1. Conclusiones

Hoy día, podemos encontrar en Internet una gran variedad de sitios dirigidos a consumidores de productos y servicios, donde, la búsqueda, selección, compra y pago de dichos productos se realiza por medios totalmente electrónicos. Pero esta forma de establecer el comercio electrónico no evita la intervención directa con el usuario, lo que puede propiciar en ciertas ocasiones que los usuarios encuentren dificultades al realizar sus compras.

En este trabajo se desarrolló un prototipo denominado STD, el cual ha sido publicado en [18]. El prototipo buscó resolver algunos de los problemas encontrados en los portales orientados a consumidores. STD se fundamenta en un modelo de tienda para comercio electrónico B2B y B2C que permite definir y enviar agentes a que realicen compras rutinarias. El prototipo está diseñado para fungir como interfaz entre empresas y usuarios.

STD permite que tanto usuarios como empresas puedan registrarse, obteniendo con ello beneficios como los que se describen a continuación. El agente encontrará las mejores opciones de búsqueda según las preferencias de los clientes. Buscará en cada una de las tiendas sin que el usuario gaste grandes cantidades de tiempo en realizar esta tarea, como lo haría tradicionalmente.

Una de las mayores ventajas que ofrece STD a las empresas proveedoras, es facilitarles una interfaz Web donde puedan ofrecer sus productos, ya sea que ellas tengan o no su propio portal Web. Sin embargo, cada empresa tiene la responsabilidad de desarrollar los servicios Web que realicen la búsqueda de los productos que ofrecen. Esta tarea actualmente puede no ser altamente factible principalmente para pequeñas empresas, a

pesar del hecho de que quizá parezca inaccesible el contar con ese tipo de tecnología, cuando es relativamente sencillo lograrla.

El prototipo está diseñado para lograr la adaptación con otros sistemas de intermediación de comercio electrónico, logrando con ello no sólo encontrar los productos que el cliente solicita dentro de las empresas registradas en STD, sino que extiende la posibilidad de acrecentar esta búsqueda hacia otras empresas quienes están registradas en el sistema de intermediación. Prueba de ello, fue la adaptación que se hizo del sistema de intermediación de comercio electrónico (desarrollado por Giner Alor Hernández) con STD.

STD buscó resolver el problema de estandarización en las categorías de productos que usan las tiendas comerciales a lo largo su desarrollo. Para lograrlo, se definió un estándar en el código de los productos que componen los 3 servidores comerciales registrados en STD. Cabe mencionar, que las empresas que soliciten registrarse dentro de STD deberán cumplir con el estándar. Este problema es actualmente difícil de resolver, debido a la diversidad de clasificaciones de productos que manejan las tiendas comerciales. Cada tienda comercial define su propio estándar, mismo que usa para la formación de los nombres, códigos y categorización de los productos que vende. Cada tienda se apega a sus intereses para la formación de este estándar. Los intereses pueden ser diversos, es por esta razón que es difícil que todas las tiendas se adapten a un estándar establecido. El logro de un estándar establecido aumentaría en gran medida las ventas en las tiendas comerciales, pese a que, para adaptarse al mismo las tiendas tendrían que modificar la forma de categorizar sus productos, situación que no es fácil de lograr.

Las contribuciones obtenidas al finalizar este trabajo de investigación son las descritas a continuación. El desarrollo de un modelo de tienda implementa un agente de búsqueda que basa su funcionamiento en servicios Web. El diseño y la creación de un portal Web que permite acceder a la funcionalidad de un sistema intermediario de comercio electrónico. Finalmente, la creación de un agente que permite buscar y comprar listas de productos en diversas tiendas usando varios criterios de selección.

6.2. Trabajo a futuro

Actualmente, los sistemas de comercio electrónico registran sitios con un alto grado de funcionalidad. Entendiéndose por este término, que los servicios que ofrecen a los usuarios son múltiples. Se pueden desarrollar un sin fin de operaciones que van desde consultar el estado del carrito de compra, hasta la elección del servicio de paquetería

para la entrega de un producto. Se propone como trabajo a futuro extender la funcionalidad de STD. En este punto, es necesario que el servicio de *entrega de productos a domicilio*, considerado fundamental para completar el ciclo de compra de productos se desarrolle. Otro de los servicios candidato a ser desarrollado es el de *manejo de preferencias del cliente* ello debido a que ha demostrado ser un servicio que ofrece alto grado de satisfacción en los clientes que lo utilizan. Se recomienda que la tecnología utilizada para el desarrollo de los servicios sea la misma usada en el agente de búsqueda, ya que se ha probado la eficiencia y rápido desarrollo logrado por el uso de la misma.

A lo largo del desarrollo del trabajo se crearon 17 servicios Web. El sistema actualmente trabaja con tres servicios. Cabe mencionar que los servicios restantes pese a comprobar su alto grado de utilidad, además del éxito logrado en sus resultados, aún no tienen funcionalidad dentro del prototipo. Se vislumbra que a futuro STD utilice estos servicios como parte de su funcionamiento básico.

De entre los aspectos de importancia encontrados, se observó la necesidad de extender los tipos de búsqueda por parte del agente. El prototipo realiza búsquedas tomando como parámetro el código de producto. Para futuras búsquedas se puede tomar la ya existente como esqueleto base para la creación de las demás. Las búsquedas por tipo de producto, empresa y por lugar de residencia se consideran los primeros candidatos a ser desarrollados, ya que se suponen mayor demandadas por los usuarios.

Se sugiere usar el sistema STD en un sitio comercial real. STD se adapta a las necesidades que actualmente demandan los sitios comerciales, en consecuencia adaptarlo a un sitio comercial es una tarea relativamente sencilla. Su uso en sitios comerciales reales reflejará un aumento en las ventas de quien lo utiliza.

Para finalizar, se pretende que STD resuelva el punto número cuatro *falta de interoperabilidad* descrito en la sección 1.3. Los beneficios que se lograrían mediante la interoperabilidad de STD con sistemas externos serían enormes. Por su diseño STD hace fácil el logro de esta tarea. p. ej. se vislumbra que a futuro STD trabaje con empresas extranjeras, las cuales manejan sus precios en dolares. STD puede interactuar con otros sistemas que le ofrezcan el tipo de cambio entre el peso y el dolar. Esto lograría que las ventas se llevarán a cabo diariamente según la cotización del dolar y el peso.

Bibliografía

- [1] A. Chavez and P. Maes. Kasbah: An agent marketplace for buying and selling goods. In *First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'96)*, pages 75–90, London, UK, 1996. Practical Application Company.
- [2] A. Moukas R. Guttman and P. Maes. Agent-mediated electronic commerce: A survey. *Communications of the ACM*, pages 1–10, 2000.
- [3] B. Faltings. Intelligent agentes: Software technology for the new millennium. *INFORMATIQUE*, pages 2–5, 2000.
- [4] N. J. Nilsson. Shakey the robot. Technical Note 323, SRI International, California, April 1984.
- [5] J. Weizenbaum. Eliza – a computer program for the study of natural language communication between man and machine. *Communications of the Association for Computing Machinery* 9, pages 36–45, January 1965.
- [6] J. Winograd. A procedural model of language understanding. In *Computers Models of Thought and Language*, pages 152–186, New York, 1973.
- [7] M. Minsky. *The Society of Mind*. Heinemann, London, 1987.
- [8] R. Guttman P. Maes and A. Moukas. Agents that buy and sell: Transforming commerce as we know it. *Communications of the ACM*, 42(3), March 1999.
- [9] Giner Alor Hernández and José Oscar Olmedo Aguirre. Sistema de intermediación para comercio electrónico b2b basado en servicios web. In *CIC*, 2003.
- [10] D. Falls and Y. Lafon. Simple object access protocol specification, soap home, <http://www.w3.org/tr/soap>, February 2004.

-
- [11] J. Jacques, R. Chinnici, M. Gudgin and S. Weerawarara. Web service description language, wsdl home, <http://www.w3.org/tr/wsdl12>, May 2004.
- [12] Microsoft Corporation. Uddi version 3.0, published, <http://uddi.microsoft.com>, July 2003.
- [13] Douglas E. Comer. *Redes de Computadoras, Internet e Interredes*. Pearson Education, 1997.
- [14] K. Tamura, H. Maruyama and N. ramote. *Creación de sitios Web con XML y Java*. Addison Wesley, 2000.
- [15] Liam Quin. World wide web consortium w3c recommendations, extensible markup language xml 1.1, <http://www.w3.org/xml>, February 2004.
- [16] R. Khalaf, W. Nagy, N. Munki, F. Curbera, M. Duftler and S. Weerawarana. Unraveling the web services web: An introduction to soap, wsdl, and uddi. *IEEE Internet Computing*, pages 86–93, March–April 2002.
- [17] Giner Alor Hernández and José Oscar Olmedo Aguirre. Automatización de la cadena de suministro en comercio electrónico usando uddi. In *CNCIICANIE*, 2003.
- [18] L. Hernández, P. Ramírez, G. Hernández, C. Sandoval and J. Olmedo. Bpim-ws: Business processes integration and monitoring for b2b e-commerce. In *11o. Congreso Internacional de Investigación en Ciencias Computacionales (CIICC'04)*, 2004.