



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN

“SISTEMA WEB PARA E-COMMERCE DE UNA TIENDA DE
CÓMPUTO. SOLUCIÓN I”

TRABAJO ESCRITO
EN LA MODALIDAD DE SEMINARIOS
Y CURSOS DE ACTUALIZACIÓN Y
CAPACITACIÓN PROFESIONAL
QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN
P R E S E N T A:
SARA VICENTE SOLANO

ASESOR: M. EN C. MARCELO PÉREZ MEDEL

MÉXICO 2005

0350985



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Objetivos del Diplomado de Desarrollo de Sistemas en Web

El presente trabajo está basado en la experiencia adquirida durante el desarrollo del Diplomado de Sistemas en Web. A continuación se describen brevemente los objetivos cubiertos durante cada módulo.

Módulo 1. Fundamentos para el desarrollo orientado al Web (www)

- Dar una vista general de la manera en que opera Internet.
- Analizar el modelo de 3 capas y justificar porqué es el recomendado para el desarrollo de un sistema web.
- Proporcionar las herramientas base para desarrollar un sistema en web.
- Exponer los aspectos a tomar en cuenta para la selección del hardware y software que permita la ejecución óptima del sistema.
- Instalar las aplicaciones para el desarrollo del sistema.
- Puntos a tomar en cuenta para plantear la propuesta del proyecto.

Módulo 2. Análisis y diseño con UML.

- Conocer y desarrollar la fase de análisis de un sistema de información mediante un enfoque orientado a objetos.
- Realizar el diseño del mismo mediante diagramas UML.
- Conocer algunas de las mejores prácticas aplicadas al desarrollo de sistemas.

Módulo 3. Introducción a la programación orientada a objetos con Java.

- Proporcionar las bases para comenzar a desarrollar aplicaciones con un enfoque orientado a objetos usando el lenguaje Java.
- Definir algunos de los estándares para la codificación de programas en Java.
- Aspectos a tomar en cuenta antes de comenzar la codificación.

Módulo 4. Capa de usuario.

A lo largo de este módulo se aprendió que las características de un prototipo están basadas en los diagramas de casos de uso y los diagramas de secuencia, lineamientos que marque el cliente en cuanto a diseño gráfico, políticas de la empresa y algunas guías de usabilidad.

El ejercicio que se practicó para esta sección fue determinar los usuarios para el sistema que se está desarrollando.

Durante el módulo se hizo un ejercicio tomando como ejemplo dos de los sistemas a desarrollar y se obtuvieron algunas ramas del mapa de navegación de ambos.

- Conocer los lineamientos para el desarrollo de una interfaz de usuario en un sistema web.
- Identificar los puntos de interacción entre el usuario y el sistema.
- Definir los niveles de usuario en base a los casos de uso y diagramas de secuencia.
- Definir el mapa de navegación del sitio web.
- Exposición de los conceptos básicos de html.
- Uso de hojas de estilo.
- Conocer la funcionalidad básica de Macromedia Dreamweaver para desarrollar la interfaz gráfica de un sitio web.

Módulo 5. Capa de datos.

- Definir la función que cumple la capa de datos de un sistema enfocado al Web.
- Entender el concepto de Gestor de Bases de Datos.
- Conocer las características de una base de datos relacional.
- Definir un esquema de una base de datos relacional.
- Aprender los comandos básicos del lenguaje SQL.
- Crear y actualizar una base de datos en DB2 y MySQL.
- Conocer algunas normas de seguridad para bases de datos.

Módulo 6. Reglas de negocio.

- Comprender que son las reglas de negocio en un sistema para el web.
- Aplicación del modelo de 3 capas.
- Publicación de una aplicación web con Jakarta Tomcat.
- Desarrollar módulos de ejemplo con la herramienta de apoyo Jakarta Struts, las librerías de Ibatis.

Introducción

En la actualidad el entorno global tan competido en el que se desarrolla la industria hace que más y más empresas se preocupen por mantener su presencia en el mercado a través de todos los canales disponibles.

Indiscutiblemente Internet se ha convertido en una posibilidad de crecimiento para varias empresas, en especial las dedicadas a la distribución de productos a usuarios finales. Este es el caso de los distribuidores de equipo de cómputo y consumibles.

Estas empresas invierten grandes cantidades de dinero en publicidad (impresos, radio, televisión, etc.) y en la última década han encontrado que un gran número de sus clientes son usuarios son asiduos visitantes de la Internet. Por eso han decidido atacar ese sector de la población económicamente activa ya que es la que mayor interés muestra en la adquisición de estos productos.

Tres de la herramientas más comunes para lograr llegar a más clientes a través de la red mundial son: envío de boletines semanales con ofertas y promociones, catálogos en línea y tiendas virtuales.

Una aplicación que combine estos tres elementos parecería ser una excelente opción para todo aquel que desee entrar al negocio de las ventas por Internet. Sin embargo esto no es suficiente, ya que se requiere de herramientas adicionales que den como resultado una solución integral a las necesidades del empresario. Esto es, administración de inventarios, actualización en tiempo real del catálogo de productos, transacciones seguras, interacción con los proveedores para poder acortar los tiempos de entrega, generación de reportes y otras herramientas para poder cuantificar la rentabilidad del sistema.

No es la finalidad de este trabajo definir las reglas del comercio electrónico, es más un intento por proponer una metodología para el desarrollo de sistema Web, basada en la experiencia adquirida durante el Diplomado de Desarrollo de Sistemas en Web. Y no sólo proponer, si no aplicar en el caso práctico de Sumicom Telemarketing S.A. de C.V. dicha metodología para demostrar que la funcionalidad del sistema sea la esperada y cubra en su totalidad las necesidades de los usuarios.

Capítulo 1

Generalidades del desarrollo en web, orientadas a un sitio de comercio electrónico.

1.1 Antecedentes

Internet es un conjunto de computadoras conectadas entre sí respetando una serie de normas establecidas en la familia de protocolos TCP/IP. No existe un mando central, todas las redes conectadas a Internet son independientes.

Una peculiaridad de esta gran red, es que, prácticamente cualquier persona alrededor del mundo puede conectarse, sólo se requiere una computadora y una cuenta de acceso con algún proveedor de servicios de Internet (PSI).

Para poder identificar a cada equipo conectado, a éste se le asigna un código numérico único que identifica a un equipo específico llamado dirección IP. Esta dirección se compone de 32 bits, esto nos da la posibilidad de tener 232 direcciones posibles. Los bits se agrupan en conjuntos de 8 bits, por cuestiones de legibilidad. Sin embargo, no es práctico el uso de números tan grandes, por lo que es mejor utilizar un nombre de dominio.

El servicio de nombres de dominio está distribuido a lo largo de Internet en varios servidores.

Las direcciones de Internet son asignadas por un organismo llamado InterNIC. El registro incluye un nombre de dominio y una dirección IP.

Ejemplo:

32 bits	11010011100110110110101011000110
1 byte	11010011.10011011.01101010.11000110
Decimal	192.168.1.0
Nombre de dominio	midominio.com

Debido a la gran diversidad de servicios que ofrece Internet (www, correo electrónico, transferencia de archivos, charlas, foros, etc.) resulta muy atractivo para las empresas llevar sus productos a la red. Y no sólo eso, también ofrecer diversos servicios se ha convertido en una prioridad.

Es por esta razón que el desarrollo de sistemas Web va en ascenso y es necesario conocer cual es la mejor arquitectura y herramientas para este tipo de sistemas.

1.2 Modelo de 3 capas

En este módulo se expusieron algunas metodologías para el desarrollo de software, por ejemplo, Tuberías y Filtros (Pipes and Filters), Capas y Gajos.

Primero que nada se necesita una arquitectura cliente-servidor, ya que nuestra aplicación residirá en un equipo remoto que será accesado por los usuarios a través de sus computadoras. Esta arquitectura nos brinda la posibilidad de trabajar por capas lógicas y permite tener las siguientes ventajas:

- Mayor flexibilidad.
- Facilidad de mantenimiento.
- Reutilizar componentes.
- Escalabilidad.

En el modelo de tres capas tenemos una capa de presentación, una con la lógica de negocios y una capa de datos como se puede ver en la figura 1.2.1.

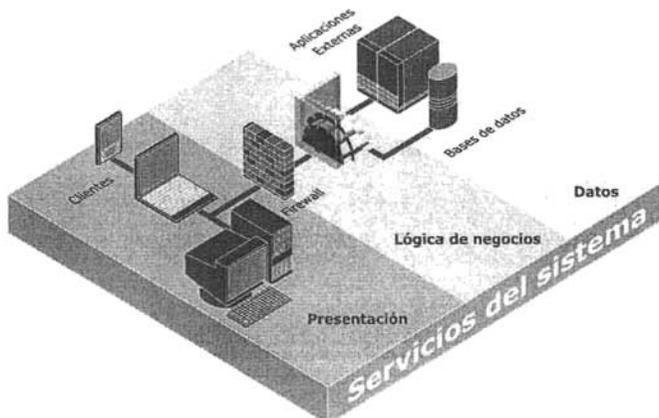


Figura 1.2.1 Modelo de tres capas

- La capa de presentación la conforman las entradas y salidas de datos en la pantalla del usuario.
- La lógica de negocios es la aplicación que procesa los datos y realiza las operaciones cruciales del sistema.
- La capa de datos es la parte de acceso a los datos, ya sean bases de datos o aplicaciones externas.

1.3 Hardware

1.3.1. *El servidor web*

Dependiendo de la aplicación serán las exigencias para el servidor web, pero en términos generales se puede hablar de una computadora con suficiente capacidad para atender numerosas peticiones de lectura a memoria y disco duro, un enlace dedicado, soporte de energía permanente y un medio ambiente adecuado para su funcionamiento óptimo (control de temperatura y humedad, ventilación, sistemas de seguridad, etc.).

1.3.2. *Estaciones de trabajo para desarrolladores*

Cada recurso involucrado en el desarrollo deberá tener un equipo de acuerdo a sus necesidades.

Por la experiencia adquirida durante este módulo, se llegó a la conclusión de que las computadoras para analistas y programadores deben ser capaces de soportar aplicaciones como Eclipse o Poseidón que sí exigen cierto grado de desempeño. Sería deseable una computadora que les permita tener movilidad.

1.4 Software

1.4.1. Sistema operativo

Al inicio de este módulo se compararon las características del sistema operativo Linux¹ contra otros sistemas y se decidió usar éste ya que ofrece las siguientes ventajas:

- Multiusuario.
- Multitarea.
- Estable.
- Manejo de un esquema de permisos.

Después de discutir las ventajas de Linux, se estudiaron sus principales características como son su estructura de directorios y archivos, permisos, usuarios, procesos y servicios, y el sistema de ventanas.

Para ver el funcionamiento del sistema operativo se proporcionó una copia de Knoppix², una de las tantas distribuciones que se pueden encontrar de este sistema operativo. Esta distribución es ejecutada desde el CD-ROM y no necesita ser instalada, utiliza el sistema de ventanas KDE y utiliza muy poco espacio en disco duro. Se llevaron a cabo algunos ejercicios en la línea de comandos como búsqueda de cadenas, reemplazos de cadenas, empaquetado y compresión de archivos, etc. En el ambiente gráfico, se examinaron algunas de las aplicaciones como procesadores de texto, navegadores, imágenes y juegos.

El siguiente paso fue llevar a cabo la instalación de otra distribución de Linux: Red Hat³, en un equipo de prueba. Las observaciones más importantes durante la instalación fueron las siguientes:

1. Antes que nada, respaldar toda la información que se tenga en el equipo en donde se va a instalar Linux. Depurar el disco duro y desfragmentarlo.
2. Tener espacio suficiente en disco para realizar dos particiones, una para la memoria virtual (SWAP) y otra para el directorio raíz.

¹ Ver <http://www.linux.org/>

² Ver <http://www.knoppix.org/>

³ Ver <http://www.redhat.com/>

3. El tamaño para la partición SWAP debe ser aproximadamente 2 veces la capacidad de memoria RAM del equipo.

Durante la instalación surgieron algunas dificultades que dieron pie a otras observaciones útiles a tomar en cuenta.

1.4.2. Servidor Web Apache

Existen diversos servidores web en el mercado, pero el más utilizado es Apache⁴ por su estabilidad, seguridad y compatibilidad con los sistemas operativos más usados. Además, al ser un proyecto abierto, está en constante desarrollo.

Para instalarlo en Red Hat se puede descargar el archivo rpm y ejecutarlo de la siguiente manera:

```
rpm -i Apache 2.###.rpm
```

Después hay que editar el archivo de configuración httpd.conf, que generalmente se encuentra en: /etc/httpd/conf/.

En Red Hat, Apache se ejecuta mediante el servicio httpd en el puerto 80; el número de puerto puede variar.

1.4.3. Secure Shell

Para poder conectarse al servidor de prueba se utilizó de una conexión segura con Secure Shell, que es un programa que encripta todos los datos que son enviados, de una manera totalmente transparente para el usuario.

1.4.4. JDK (J2SE Development Kit)⁵

Es un conjunto de herramientas para el desarrollo de aplicaciones en Java. Durante este módulo, se descargó y se instaló el JDK, y se hicieron las modificaciones necesarias al las variables de entorno.

⁴ Ver <http://www.apache.org/>

⁵ Ver <http://java.sun.com/j2se/>

1.4.5. Servidor de aplicaciones Tomcat

Para la implementación de páginas dinámicas se necesita un servidor de aplicaciones, durante el diplomado se usará Java. Por esta razón se eligió Tomcat⁶ que es un contenedor de servlets. Al igual que Apache es un proyecto abierto mantenido por desarrolladores de todo el mundo.

Una vez puesto a punto el servidor se compiló una aplicación de ejemplo.

1.4.6. Manejador de bases de datos

Para todo sistema basado en el modelo de tres capas, se necesita una aplicación que provea de las herramientas necesarias para el manejo de los datos. A estas aplicaciones se les conoce como manejadores de bases de datos.

Durante este módulo se dio una breve introducción a los conceptos básicos de bases de datos relacionales y sentencias SQL (Structured Query Language).

El manejador de bases de datos seleccionado para este diplomado es mySQL⁷, un manejador muy popular, pero no tan robusto como otros manejadores, por ejemplo Oracle. Sin embargo, muchos sistemas web actualmente usan mySQL como su gestor de datos principal.

1.5 Propuesta del proyecto

El último punto que se abarcó en el módulo fue el desarrollo de la propuesta del proyecto. Desde un principio se asignó la tarea de elegir un problema a resolver mediante un sistema web, para al final del módulo, entregar una propuesta.

Los puntos a tomar en cuenta para la propuesta son:

- Descripción de la organización en cuestión.
- Requerimientos del Sistema.
- Definición del Problema.
- Back Office.

⁶ Ver <http://jakarta.apache.org/tomcat/>

⁷ Ver <http://www.mysql.com/>

- Quiénes intervienen.
- Tareas.

1.5.1. Generalidades

Sumicom Telemarketing, S.A. de C.V. (a quien de ahora en adelante se llamará Sumitel simplemente) empresa líder en venta de artículos de cómputo realiza la mayoría de sus actividades sin automatización alguna, las ventas sólo las realiza vía telefónica y la administración de las mismas así como del inventario y otros recursos quedan sin registro electrónico.

El sistema de ventas por teléfono de Sumitel se ha saturado y necesita ser reforzado para llevar a cabo una labor de venta más eficiente.

El mercado principal de Sumitel, actualmente se limita a la zona centro del País y a las zonas donde se tiene sucursales (Toluca, Guadalajara, Morelia, San Luis Potosí).

El tiempo transcurrido desde el levantamiento de pedido hasta la recepción del mismo en el domicilio del cliente, es largo y la única forma de pago posible es el depósito bancario.

Esta problemática se ha mantenido vigente debido a que el cliente no encontraba una solución que se amoldara a sus expectativas y presupuesto.

Los resultados que se obtendrán al aplicar esta solución serán tangibles y entre ellos podemos mencionar:

- ✓ Se reducirá la carga de trabajo del equipo de tele marketing mediante la implementación del sitio web para comercio electrónico.
- ✓ Generar mayores ventas llegando a más usuarios en la República Mexicana.
- ✓ Se proporcionará un sistema de venta sencillo y a la vez atractivo al cliente, lo que generará más visitas al sitio web.
- ✓ Se proporcionará un medio seguro para la adquisición de equipo de cómputo a través del pago electrónico con tarjeta de crédito o débito.
- ✓ Se optimizarán los tiempos de entrega.

El usuario de sistemas de comercio electrónico es muy exigente y si no se cumplen sus expectativas, optará por visitar cualquier otro sitio, tomando como base tal planteamiento se ofrecen como ventajas:

- ✓ Facilidad de uso y exploración
- ✓ Rápido rendimiento
- ✓ Compras anónimas
- ✓ Perfil de usuario
- ✓ Seguridad eficaz
- ✓ Competitividad gracias a la facilidad de uso.

1.5.2. *Facilidad de uso y exploración*

Un sitio Web debe ofrecer facilidad de uso y exploración. Después de todo, las empresas desean facilitar la compra de sus productos y es más probable que el cliente emplee su dinero en un sitio en el que pueda localizar fácilmente la página de pedidos.

Una forma de facilitar la utilización del sitio consiste en emplear analogías conocidas para las tareas comunes. Esto significa, por ejemplo, que las selecciones del cliente se puedan almacenar en un carrito de compra hasta que se complete (o se efectúe un *pedido*). El uso de metáforas como ésta facilita que los usuarios que no poseen un amplio conocimiento de informática comprendan el funcionamiento del sitio y puedan efectuar sus compras.

Facilitar la exploración del sitio resulta una tarea mucho más compleja de lo que pueda parecer en un primer momento. El Web funciona de un modo bastante poco lineal y los usuarios suelen hacer clic en los vínculos sin seguir un orden determinado. Es necesario que el sitio disponga de una interfaz sólida para el usuario, independientemente del tipo de página que se esté consultando en ese momento; asimismo, las páginas de mayor importancia (como, por ejemplo, la página principal, el carrito de compras y la información de la cuenta del usuario) deben encontrarse siempre disponibles con sólo hacer clic en un vínculo. Otra manera de asegurar un fácil manejo del sitio consiste en organizar la lista de productos y catálogos de forma lógica. Si los catálogos se dividen en categorías o

en varias subcategorías, el cliente podrá localizar fácilmente el producto que desea. Asimismo, se debe ofrecer la posibilidad de buscar un producto si no se sabe con seguridad dónde se incluye el mismo.

Si el sitio ofrece facilidad de uso y exploración, los clientes disfrutarán utilizándolo. De lo contrario, si presenta dificultades, probablemente abandonen el sitio y opten por visitar otro distinto.

1.5.3. Rápido rendimiento

Existen varios factores en el diseño de un sitio Web que pueden afectar a su rendimiento. Es obvio que el rendimiento adquiere distintos significados dependiendo del usuario y que los niveles de rendimiento que se consideran aceptables pueden variar de un usuario a otro.

1.5.4. Reducción del tiempo de respuesta

El factor que se considera como el más indicativo del buen funcionamiento de un sitio suele ser que se proporcione un tiempo de respuesta aceptable. El tiempo de respuesta es la cantidad de tiempo de espera que transcurre desde que se realiza una petición hasta que se obtiene el resultado. Lo ideal sería que todas las acciones del sitio se realizaran de forma instantánea; sin embargo, en la práctica, factores como un ancho de banda limitado, la concurrencia a la base de datos y las tareas de procesamiento empresarial suelen ser la causa de ligeros retrasos. Uno de los objetivos del proceso de diseño de un sitio de comercio electrónico debe consistir en intentar reducir los factores que afectan negativamente al tiempo de respuesta, aunque dichos factores no se puedan eliminar en su totalidad.

La clave para la optimización del comercio electrónico consiste precisamente en optimizar el tiempo que conlleva realizar operaciones tales como la realización de pedidos para que, de este modo, éstos no se pierdan por el hecho de que los clientes abandonen sus carritos de compras mientras transcurre el tiempo de espera.

1.5.5. Compras anónimas (Exploración)

En general, el usuario no desea tener que iniciar la sesión sin ver antes los artículos de venta que se encuentran disponibles en el sitio. El sitio debe permitir que los usuarios anónimos exploren los productos e incluso que sitúen los artículos en un carrito de compra, sin que se les exija una autenticación previa.

1.5.6. Perfil de usuario

Cuando un cliente vuelve a visitar un sitio, no desea tener que volver a introducir los mismos datos que introdujo en la visita anterior. Una vez proporcionada la información de contacto y de envío, el usuario espera que estos datos se recuerden.

Para llevar a cabo esta operación, muchos sitios conservan la información de *perfil* de usuario de todos los clientes que se registran en el sitio. En la mayoría de los casos, se pide al usuario que realice el registro proporcionando una cantidad mínima de información del perfil como, por ejemplo, la contraseña y nombre de usuario. Posteriormente, al usuario se le asigna un identificador único que se puede utilizar como clave principal para los datos de su perfil.

Tras el registro en el sitio, la información del perfil se puede guardar en una base de datos y recuperarse para su uso posterior. Generalmente, el usuario puede agregar la información obligatoria, especificando detalles como, por ejemplo, la dirección de correo electrónico, el número de teléfono, la dirección de envío o cualquier otra información personal que se pueda agregar.

El hecho de conservar la información del perfil de usuario resulta útil por las siguientes razones:

- Evita que el usuario tenga que volver a introducir los datos en visitas posteriores.
- Se puede emplear para analizar la actividad del usuario en el sitio.
- Se puede utilizar como sistema de personalización, permitiendo dirigir titulares o descuentos a un tipo específico de usuario.

- Se puede utilizar con motivos de análisis comercial como, por ejemplo, averiguar las tendencias de compra basándose en unos valores de perfil determinados.

1.5.7 Competitividad gracias a la facilidad de uso

Los diseñadores de la aplicación no son responsables de decisiones comerciales tales como los precios, las campañas publicitarias, etc. No obstante, el diseño de una solución de comercio electrónico puede tener una gran influencia en la forma en la que un negocio puede reaccionar frente a las tendencias del mercado y la actividad de la competencia.

1.6 Características particulares del proyecto

1.6.1 ¿Qué problema resuelve este proyecto?

El sistema de ventas por teléfono de Sumicom Telemarketing S.A. de C.V. se ha saturado y necesita ser reforzado para llevar a cabo una labor de venta más eficiente.

El mercado principal de Sumitel, actualmente se limita a la zona centro del país y a las zonas en donde tiene sucursales (Toluca, Guadalajara, Morelia, San Luis Potosí). Esporádicamente se realizan ventas a otros estados.

El tiempo transcurrido desde el levantamiento del pedido hasta la recepción del mismo en el domicilio del usuario, es largo, ya que la única forma de pago es por depósito bancario.

1.6.2 Objetivo del proyecto

Reducir la carga de trabajo al equipo de tele marketing mediante la implementación de un sitio web para comercio electrónico. Esto permitirá generar mayores ventas llegando a más usuarios en toda la república mexicana.

Hacer que la experiencia de compra a través del sitio sea atractiva y, a la vez, sencilla, provocando que el usuario regrese.

Al proporcionar un medio seguro para la adquisición de equipo de cómputo a través del pago electrónico con tarjeta de crédito o débito, los tiempos de entrega serán menores.

Ofrecer mejor calidad en el servicio al proporcionar herramientas que permitan dar seguimiento a los pedidos y mantener al día a los usuarios mediante el envío de boletines electrónicos.

1.6.3 ¿Cuál es el alcance de este proyecto?

Se desarrollará un sistema web para comercio electrónico que comprende dos subsistemas:

1. Sitio para venta directa de equipo de cómputo (tienda).
2. Sitio de administración para control y actualización de la tienda.

1.6.4 Objetivos Acordados

Después de revisar los documentos enviados por la empresa y haber tenido una reunión con los encargados del departamento de ventas, se definieron las características principales que el sistema debe cubrir.

1.6.5 Ambiente

¿Cuál es el ambiente de negocios del sistema?

El usuario busca un equipo de cómputo que se adapte a sus necesidades y presupuesto. Por esta razón recorre varios sitios web que le ofrecen diversas marcas y precios.

Se inclinará más a comprar en un sitio que, de manera fácil y segura, le permita conocer a detalle varias opciones en equipo de cómputo.

Este sitio deber permitirle todo lo anterior y brindarle un valor agregado, como pueden ser descargas gratuitas de controladores y utilerías, además de artículos de interés y noticias.

Se debe crear un vínculo estrecho entre proveedor y consumidor a través de un servicio personalizado y confiable.

¿Cuál es el ambiente físico del sistema?

El servidor web en donde se alojará el sistema se encuentra en las instalaciones del proveedor del servicio, en este caso, nuestrosite.com.

El servidor cuenta con respaldo de energía permanente, soporte técnico 24x7 y temperatura controlada.

¿Cuál es el ambiente tecnológico del sistema?

El usuario promedio que visita los sitios de venta de computadoras cuenta con un equipo con las siguientes características:

- Computadora personal
- Procesador: Pentium II
- Monitor 15"
- Resolución de pantalla: 800 X 600
- Sistema operativo: Windows XP
- Explorador principal: Internet Explorer

1.6.6 Back office

- Gerencia
- Equipo de ventas
- Almacén
- Manejo y embalaje de paquetes

1.6.7 Actores y Funciones

- Gerente

Responsable directo de la tienda.

Debe tener asignada una cuenta de usuario en el sistema y una contraseña para poder acceder al panel de administración.

Únicamente podrá consultar reportes generados por el sistema. Por ejemplo: reportes de ventas, de clientes registrados recientemente, compras, datos de proveedores, etc.

- Administrador

Responsable de la contabilidad y administración de la tienda. Consulta de stock y reportes.

Debe tener asignada una cuenta de usuario en el sistema y una contraseña para poder acceder al panel de administración.

- Vendedor

Captura, cancela, modifica pedidos. Ejecuta la venta.

Debe tener asignada una cuenta de usuario en el sistema y una contraseña para poder acceder al panel de administración.

- Encargado del catálogo de productos.

Mantiene actualizado el catálogo e inventario de productos.

Debe tener asignada una cuenta de usuario en el sistema y una contraseña para poder acceder al panel de administración.

- Encargado de bodega

Administra stock de productos (entradas y salidas).

Debe tener asignada una cuenta de usuario en el sistema y una contraseña para poder acceder al panel de administración.

- Despachador

Prepara paquetes para envío por paquetería o para venta en mostrador.

Debe tener asignada una cuenta de usuario en el sistema y una contraseña para poder acceder al panel de administración.

- Cliente

Desea encontrar rápidamente los productos de interés.

Una vez encontrado, desea obtener la información más importante de ese producto, pero también debe tener opción para ver más detalles como hojas técnicas, manuales, controladores, presentaciones del fabricante, etc.

Puede navegar anónimamente en el sitio para conocer los productos. Solamente necesitará autenticación cuando vaya a concretar una compra, cuando vaya a agregar productos favoritos, actualizan sus datos, si es que ya está registrado, crear un boleto de ayuda, etc. (El punto de boletos de ayuda aún está por definirse).

- Administrador del sistema

Es el encargado de administrar todo el sistema.

Debe tener asignada una cuenta de usuario en el sistema y una contraseña para poder acceder al panel de administración.

Su tarea es crear, dar de baja y modificar usuarios. Es el único con los privilegios necesarios para eliminar permanentemente registros del sistema (sólo de manera lógica, la eliminación física no es posible).

1.6.8 Panorama actual

Ventas

En mostrador:

4. Cliente llega a la tienda, verifica y selecciona productos
5. Vendedor levanta pedido
6. Se verifican existencias en bodega
7. Se surte el pedido
8. Se empaca
9. Se entrega al cliente o se envía a la compañía de paquetería

Por teléfono:

1. Cliente tiene catálogo impreso de productas, selecciona
2. Llama a la tienda
3. Vendedor levanta pedido
4. Se verifican existencias en bodega
5. Se surte el pedido
6. Se empaca
7. Se envía a la compañía de paquetería

1.6.9 Panorama Automatizado

Ventas

En mostrador:

1. Cliente llega a la tienda, verifica y selecciona productos
2. Vendedor verifica existencias en línea
3. Vendedor levanta pedido
4. Se surte el pedido
5. Se empaca
6. Se entrega al cliente o se envía a la compañía de paquetería
7. Cliente verifica estado de su pedido en línea hasta que lo recibe

Por teléfono:

1. Cliente verifica catálogo
2. Llama a la tienda
3. Vendedor verifica existencias en línea
4. Vendedor levanta pedido
5. Se surte el pedido

6. Se empaca
7. Se envía a la compañía de paquetería
8. Cliente verifica estatus de su pedido en línea hasta su recepción

Y se agrega una opción más de comercialización para los clientes y prospectos

En línea:

1. Cliente checa catálogo en línea
2. Agrega productos a su carrito de compras
3. Ingresa al sistema
4. Procesa su pedido
5. Se recibo pedido en la sucursal
6. Se checa existencias en bodega
7. Se surte el pedido
8. Se empaca
9. Se envía a la compañía de paquetería
10. Cliente verifica estatus de su pedido en línea hasta su recepción

1.6.10 Tareas Nuevas

Venta en línea

1. Compra vía Internet
 - a. Ingreso al sitio web
 - b. Buscar productos en catálogo
 - c. Agregar productos a carrito de compras
 - d. Procesar pedido
 - e. Registrar cliente

- f. Confirmar pedido
- g. Ingresar pago
- h. Verificar pago
- i. Procesar Pago
- j. Crear pedido
- k. Verificar pedido
- l. Checar stock
- m. Surtir pedido
- n. Enviar pedido

1.6.11 Expectativas

Se pretende generar mayores ventas a través de la tienda virtual llegando a más usuarios en toda la república mexicana. Además se ofrece mejor calidad en el servicio al tener la posibilidad de dar seguimiento a los pedidos de cada cliente.

La empresa desea tener diferentes grupos de clientes a los cuales se les pueda aplicar descuentos, ofrecerles listas de precios especiales y promociones exclusivas. Los grupos serán creados sobre la base del volumen de compras o la frecuencia.

Con la finalidad de ponerlo en un nivel competitivo y automatizar sus procesos, se sugiere la implementación de un sitio web dividido en tres partes:

- Panel de administración.
- Interfaz para clientes (tienda).
- Interfaz para visitantes.

En términos generales el objetivo del sitio web es el dar a conocer y vender los productos que ofrece la empresa y llegar a un mayor número de clientes potenciales a nivel nacional. Además de establecer una relación más estrecha con proveedores lo cual se debe reflejar en tiempos de entrega más cortos.

En resumen destacaremos los siguientes puntos

1. Proporcionar un sitio en el que se permita al cliente conocer los productos que la empresa comercializa (Catálogo).
2. Brindarle una herramienta para realizar sus pedidos de manera sencilla (Tienda).
3. Conocer de manera más específica al público que visita el sitio, en base a preferencias.
4. Módulo de registro de clientes.
5. Módulo de registro de proveedores.
6. Manejo de cuentas por visitantes, clientes y proveedores.

1.7 Alcance

Queremos enfocarnos en el diseño de una aplicación web con capacidad de manejar ventas en línea y realizar transacciones de forma confiable. Esto con el propósito de contar con un servicio configurable, amigable y que se mantenga vigente con el tiempo y con el crecimiento.

En alcance	Fuera del alcance
Construir una aplicación que pueda usarse con servidores estándares de Web y de aplicación	Construir un nuevo servidor de aplicaciones Web
Hacer que la interfaz sea compatible con los navegadores IE y Mozilla	Mantener compatibilidad con navegadores poco comunes o de versiones muy atrasadas
Seguridad en el sistema a través de usuarios/contraseñas y niveles de acceso	Protección total de la aplicación contra ataques de "Hackers".
Una interfaz de usuario sencilla y amigable	Una lista sofisticada de "skins" para adaptar la interfase de usuario a cada usuario en particular

1.8 Entregables

- Aplicación web para “Comercio Electrónico Caso Práctico Sumitel”.
- Muestra de la interfaz de usuario.
- Documentación UML avalando el proyecto.

1.9 Riesgos y beneficios

1.9.1 Riesgos

Existe un conflicto potencial entre el objetivo de diseñar un motor adaptable a distintos negocios del mismo giro y que éste sea fácilmente configurable a las necesidades cambiantes del cliente.

1.9.2 Beneficios

Si cumplimos el alcance del sistema nuestra aplicación dará un beneficio en ahorro operativo y mejora en calidad de servicio a quienes no quieren emplear tiempo en trasladarse y comprar directamente en tienda.

1.10 Propuesta técnica

Los puntos anteriormente abarcados se han resumido en el siguiente documento que será entregado al cliente.

Propuesta Técnica para desarrollo del sistema web, caso práctico Sumitel.

Introducción

El presente documento pone a su consideración la propuesta técnica y funcional de un **SISTEMA WEB PARA E-COMMERCE DE UNA TIENDA DE CÓMPUTO** que cubra las necesidades de venta y promoción del cliente **Sumicom Telemarketing S.A. de C.V.**, quien tiene como giro la venta directa de computadoras y componentes, así como servicios de mantenimiento preventivo y correctivo a equipo de cómputo.

Esto a través del análisis óptimo y de un plan de actividades encaminadas a lograr una implementación exitosa.

Objetivos del Documento

El alcance establecido en la propuesta se enfoca en las características y funciones fundamentales del cliente.

Objetivos del Proyecto

En base a los requerimientos solicitados al cliente se sugiere la implementación de un sitio web, con la finalidad de poner al cliente a un nivel competitivo y automatizar sus procesos, el cual está dividido en tres partes:

- Administración.**
- Clientes.**
- Visitantes.**

En términos generales el objetivo del sitio web es el dar a conocer y vender los productos que ofrece la empresa y llegar a un mayor número de clientes potenciales a nivel nacional. Además de establecer una relación más estrecha con proveedores lo cual se debe reflejar en tiempos de entrega más cortos.

En resumen destacaremos los siguientes puntos:

10. Proporcionar un sitio en el que se permita al cliente conocer los productos que la empresa comercializa (Catálogo).
11. Brindarle una herramienta para realizar sus pedidos de manera sencilla (Tienda).
12. Conocer de manera más específica al público que visita el sitio, en base a preferencias.
13. Módulo de registro de clientes.
14. Módulo de registro de proveedores.
15. Manejo de cuentas por visitantes, clientes y proveedores.

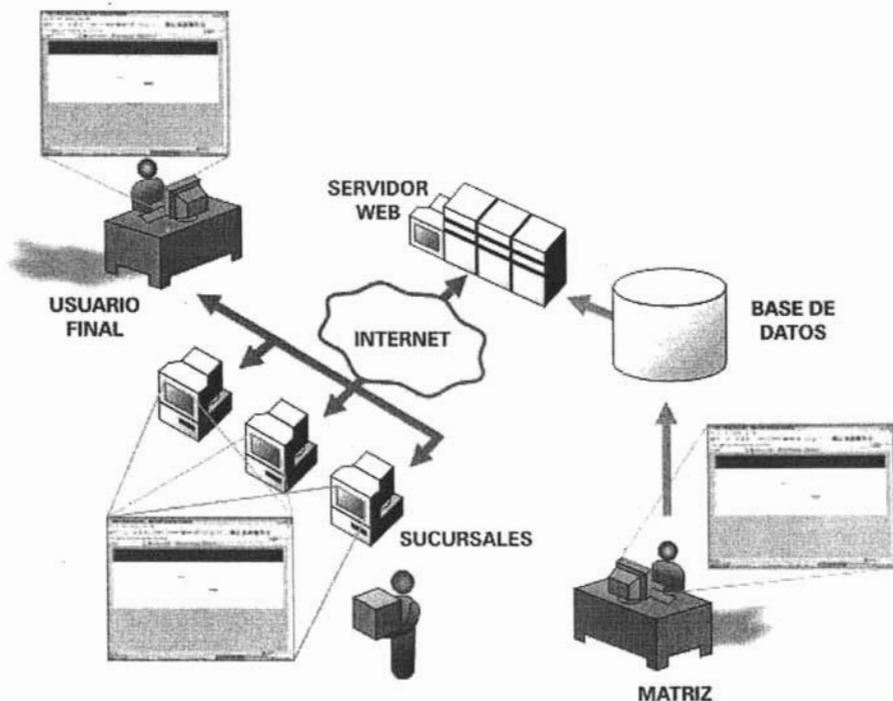
Características del Sistema

Se propone el uso de un panel de control para la administración de los catálogos de productos, clientes, proveedores y el inventario de la tienda.

El acceso a estos paneles estará protegido por nombres de usuario y contraseñas, además de restricciones por horario y días de la semana. Estos permisos serán asignados por el administrador global del sistema.

Se considera, además, la generación de un conjunto de reportes impresos con información resumida y clasificada, según especificaciones del cliente.

Toda la información será almacenada en una base de datos central y podrá ser consultada vía Internet en tiempo real, a través de un sitio web que representa la tienda virtual en sí.



1.10.1 Logística del sitio web.

El sistema web propuesto consta de la siguiente estructura:

Tienda virtual

1. Página Principal
Contenido por definir.
2. Menú de productos
Categorías definidas previamente.
3. Búsqueda de productos
En base a criterios definidos previamente.
4. Productos
Listado y ficha de detalles de los productos comercializados por la empresa.

5. Carrito de compras

El cliente agrega o elimina los productos de su interés. Se hace un cálculo del total y subtotal sin impuestos ni cargos por gastos de envío.

6. Confirmación de pedidos

Si el cliente está registrado, puede procesar su pedido, o sea, confirmar la compra de los productos en su carrito. Si no está registrado deberá darse de alta antes de continuar.

7. Registro de clientes

El cliente proporcionará sus datos, los cuales serán definidos por la empresa previamente.

8. Registro de visitantes

Los visitantes del sitio proporcionarán sus datos y preferencias que se seleccionarán en una lista de opciones.

9. Mi cuenta

a. Perfil

El cliente podrá modificar sus datos en esta sección.

b. Pedidos

Consulta de pedidos exclusiva para clientes (estado, importe, cancelaciones).

10. Políticas de Privacidad

Por definir.

11. Términos y Condiciones de Uso

Por definir.

Panel de Administración

1. Página Principal
Contenido por definir.
2. Productos
Administración del catálogo.
3. Clientes
Administración de clientes.
4. Pedidos
Administración de pedidos generados en la tienda o manualmente desde el panel.
5. Proveedores
Administración de proveedores.
6. Compras
Administración de pedidos a proveedores.
7. Inventario
8. Estadísticas
Resumen de datos de los visitantes registrados, ordenados en base a criterios especificados (edad, sexo, preferencias, etc.).
9. Utilerías
Herramientas para mantenimiento de bases de datos.

Características Técnicas

El sistema propuesto estará programado bajo los estándares de código abierto.

Sistema Operativo: Red Hat Linux.

Servidor Web: Apache 2.0.

Contenedor de Servlets: Apache Tomcat 5.5.9.

Lenguajes de programación: Java, Javascript, Actionsript.

Manejador de Bases de Datos: MySQL.

Tiempo de Entrega

8 semanas

Plan Detallado de Actividades

Gráfica de Gantt

Actividad	SEMANAS												
	1	2	3	4	5	6	7	8	9	10	11	12	
Levantamiento de requerimientos del cliente	■												
Dimensionamiento	■												
Análisis y Diseño		■	■										
Creación de bases de datos				■									
Prototipo					■								
Diseño de interfaz gráfica					■	■							
Programación						■	■	■					
Implementación									■				
Pruebas										■			
Control de cambios											■		
Liberación												■	

Estos tiempos son estimados considerando que el cliente ha proporcionado ya toda la información y material necesario para el desarrollo del sistema web.

Capítulo 2

Análisis UML para sitio Web

2.1 Generalidades

Cualquier sistema informático debe contar con tres características básicas: escalabilidad, seguridad y una ejecución robusta bajo condiciones críticas.

Su estructura debe ser modular para que la detección y corrección de fallas se haga de la manera más rápida posible.

Para lograr esto es necesario que la etapa del diseño del sistema sea la más exhaustiva. Si bien la etapa de programación es crucial, un buen diseño puede economizar tiempo y recursos a lo largo del desarrollo del proyecto.

La localización temprana de módulos y sus relaciones entre si permite que las librerías creadas para cada módulo puedan ser utilizadas nuevamente en un futuro.

Esto se refleja directamente en el costo del proyecto si tomamos en cuenta que para cada actualización no habrá necesidad de programar toda la aplicación.

El uso de estándares es otro aspecto importante a tomar en cuenta. Cuando la arquitectura de una aplicación está basada en estándares, su desarrollo será más fluido.

Además es más fácil que el proyecto sea retomado por nuevos desarrolladores si en alguna etapa fuera necesario integrar nuevos recursos.

Entre más grande sea un sistema, más compleja será la etapa de diseño y muchas más las posibilidades de que este sistema fracase. En la vida práctica, muchos de los sistemas grandes nunca son liberados debido a su inviabilidad ya sea porque sobrepasan el presupuesto asignado o porque simplemente no cumplen los requerimientos del cliente. En este caso un modelo permite tener una perspectiva más clara del problema, como resolverlo, crear una solución y cómo integrar esta solución para que al final pueda ser traducida en líneas de código.

Un sistema con un buen análisis es más fácil de mantener y de escalar.

Existen diversos enfoques para el modelado de un sistema, los tradicionales y el orientado a objetos.

El modelado orientado a objetos permite cubrir con muchos de los puntos anteriormente mencionados.

Una de las herramientas más populares para el modelado de objetos es UML¹ (Unified Modeling Language).

UML es una especificación mantenida y definida por el OMG (Object Management Group), que se encarga de gestionar las especificaciones para la industria de la computación sin fines de lucro.

A lo largo de este módulo se cubrieron los puntos más importantes para el análisis y diseño de un sistema, con un enfoque orientado a objetos. Se realizaron diversos ejercicios con los diagramas UML más utilizados actualmente.

2.2 Análisis

Al inicio de este módulo se remarcó la importancia de la etapa de análisis en el proceso de desarrollo de cualquier sistema. Siempre se debe hacer un análisis antes de comenzar a codificar, esa es una regla de oro.

Un buen análisis lleva a tomar una buena decisión, lo cual deriva en el desarrollo de una aplicación exitosa.

El sistema que se tomó como ejemplo durante el módulo 2 es el de una empresa promotora de espectáculos de diversos géneros que desea vender boletos por medio de su sitio web.

2.3 Ciclo de vida de un sistema

2.3.1. Encuesta

Lo primero que necesitamos para comenzar el análisis de nuestro sistema es conocer las necesidades del usuario. Una de las mejores maneras es hacerlo por escrito a través de una encuesta o cuestionario; ahí se expresarán todas las dudas que tenga el o los analistas acerca de la funcionalidad del sistema. Este cuestionario se entrega al usuario para ser contestado por escrito. Una de las finalidades de este documento es tener un punto de partida para determinar el alcance del proyecto y además, implica un compromiso entre el equipo de

¹ UML™ Resource Page (<http://www.uml.org>)

desarrollo y el usuario, es por esta razón que debe ser redactado de manera formal.

Una vez conocidas las necesidades del usuario se puede comenzar formalmente la etapa de análisis que puede ser visto desde dos puntos de vista: del entorno y del sistema en sí.

2.3.2. *Hardware y software*

Físicamente en dónde se instalará la aplicación y que software será necesario para que el desempeño sea el óptimo. Con base a este análisis se decidirá si se necesita comprar hardware y software nuevo, o bien, se puede utilizar el que se tiene actualmente.

2.3.3. *Factibilidad*

Éste es un punto muy importante, es aquí en donde se decide si conviene el desarrollo del sistema, tanto al usuario como al desarrollador.

2.3.4. *Estimación*

Tomar en cuenta el margen de error y dejarlo en claro para ambas partes.

2.3.5. *Análisis de Costo-Beneficio*

Va de la mano con el punto de factibilidad, el usuario decidirá apeguándose a los resultados de este análisis, si le conviene invertir en el proyecto.

La etapa de análisis permite tomar decisiones que convengan tanto al usuario como a nosotros como desarrolladores.

Si el usuario considera seguir adelante con el proyecto, podemos pasar a la etapa de diseño.

2.4 **Diseño**

El primer documento que podemos obtener en la etapa de diseño es la especificación de procesos; debe ser redactado de manera clara ya que este será el antecedente de los casos de uso.

El siguiente paso es identificar entidades y relaciones, y representarlas gráficamente en un diagrama.

Debemos entender por entidad un objeto del mundo real que tiene características propias que lo definen y distinguen de otros. No obstante, los objetos pueden tener atributos comunes, esto crea una relación entre ellos.

El diagrama de entidad relación debe ir siempre acompañado de un diccionario de datos, que es un compendio de los atributos de cada entidad en donde se define nombre del atributo o campo, características y tipo de dato.

Algunas recomendaciones a tomar en cuenta al elaborar el diagrama de entidad relación son:

- Identificar el mayor número de entidades involucradas en el sistema.
- Encontrar el mayor número de atributos para cada entidad.
- Intentar relacionar el mayor número posible de entidades del sistema.
- Es deseable el manejo de atributos numéricos.
- El nombre de las entidades debe ser significativo estar dado en singular.
- Al nombrar los atributos, asignarles un prefijo asociado con el nombre de la entidad.
- Si una entidad tiene más de un valor posible para uno de sus atributos, ese atributo debe ser una entidad.
- Mantener siempre un campo que indique el estatus del registro (vigente, eliminado, etc.).

Una entidad puede ser vista como una tabla de una base de datos, todas las tablas están formadas por registros que, a su vez, están compuestos por campos.

Las entidades deben tener un campo o conjunto de campos que permitan garantizar que cada registro es único, estos campos se conocen como llaves primarias. Cuando una entidad se relaciona con otra, los campos que forman la llave primaria pasan a formar parte de la otra entidad y se conoce como llave foránea. Las llaves primarias formadas por varios campos no pueden separarse al pasar a otra entidad.

Una llave foránea no necesariamente debe formar parte de la llave primaria de la tabla con la que se está relacionando.

El producto que se obtiene de la fase de diseño de un sistema es un modelo.

2.4.1 Diccionario de datos

En esta etapa de identificaron las entidades para el sistema propuesto, a continuación se muestra el diccionario de datos obtenido.

Entidad:	Pedido	
Descripción:	Almacenar toda la información relativa a los pedidos de los clientes.	
Número de ejemplares:	-	
Crecimiento previsto:	80 pedidos/mes.	
Confidencialidad:	Los datos podrán ser consultados sólo por el cliente al que corresponde el pedido, el vendedor que levantó el pedido, el encargado del almacén o el administrador del sistema. Los datos correspondientes a la forma de pago sólo podrán ser consultados por el cliente mismo, el vendedor y el administrador.	
Derechos de acceso:	Las consultas serán previa autorización de nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
ped_id	Identificador del pedido.	Int(10)
cli_id	Identificador del cliente al que pertenece el pedido.	Int(10)
ped_fecha	Fecha en que se realizó el pedido dd-mm-aaaa hh:mm:ss.	Date
ped_fecha_ent	Fecha de entrega del pedido dd-mm-aaaa hh:mm:ss.	Date
ped_status	Código del estatus del pedido, aún por definir.	Int(2)
dom_id	Identificador del domicilio fiscal.	Int(10)

Entidad:	Detalle_Pedido	
Descripción	Partidas de los pedidos.	
Número de ejemplares:	-	
Crecimiento previsto:	-	
Confidencialidad:	Los mismos que para los pedidos.	
Derechos de acceso:	Las consultas serán previa autorización de nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
Ped_cve	Identificador del pedido detallado.	Int(10)
Prod_id	Identificador de cada producto que compone el pedido.	Int
Dp_cantidad	Cantidad de cada producto.	Int
Prod_precio	Precio unitario de cada producto.	Float
Dp_desc	Descuento sobre el producto.	Float
Dp_iva	IVA del pedido.	Float
Dp_tot	Importe total.	Float

Entidad:	Envío	
Descripción	Almacena la información referente al envío de los pedidos.	
Número de ejemplares:	-	
Crecimiento previsto:	-	
Confidencialidad:	Los mismos que para los pedidos.	
Derechos de acceso:	Las consultas serán previa autorización de nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
Env_num	Identificador del comprobante de envío.	Int
Env_fechaenv	Fecha en la que fue enviado el pedido. dd-mm-aaaa hh:mm:ss.	String
Env_fechaec	Fecha en la que fue recibido el pedido. dd-mm-aaaa hh:mm:ss.	String
Env_remitente	Remitente.	String
Env_destinatario	Destinatario.	String
Env_status	Clave del Estatus del envío (por definir).	Int
Mens_id	Identificador de la compañía de mensajería.	Int
Env_peso	Peso del envío.	Float
Ped_cve	Identificador del pedido que se está enviando.	Int

Entidad:	Producto	
Descripción:	Almacenar toda la información relativa a los productos.	
Número de ejemplares:	228.	
Crecimiento previsto:	5 productos/mes.	
Confidencialidad:	Los datos podrán ser consultados por cualquier persona. La modificación de productos sólo puede realizarla el encargado del catálogo y el administrador.	
Derechos de acceso:	La actualización será previa autorización con nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
Cat_id	Identificador de la categoría.	Int
Scat_id	Identificador de la subcategoría.	Int
Prod_id	Identificador del producto.	Int
Prod_desc	Descripción del producto.	String
Prod_precio	Precio base de este producto.	Float
Prod_marca	Identificador de la marca de este producto.	Int
Prod_rutaimgchica	Ruta de la imagen de muestra del producto.	String
Prod_rutaimggrande	Ruta de la imagen del producto.	String

Entidad:	PropiedadProducto	
Descripción:	Almacenar las propiedades de un producto.	
Número de ejemplares:	3 por producto capturado.	
Crecimiento previsto:	Depende directamente de los productos.	
Confidencialidad:	Los datos podrán ser consultados por cualquier persona, siempre y cuando se consulte un producto. La modificación de propiedades sólo puede realizarla el encargado del catálogo y el administrador.	
Derechos de acceso:	La actualización será previa autorización con nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
Prop_id	Identificador único de esta propiedad.	Int
Prod_id	Identificador del producto al que describe esta propiedad.	Int
Prop_propiedad	Nombre de la propiedad.	String
Prop_valor	Valor de la propiedad.	String

Entidad:	Categoría	
Descripción:	Almacenar toda la información relativa a las categorías de los productos.	
Número de ejemplares:	9	
Crecimiento previsto:	1 categoría/año	
Confidencialidad:	Los datos podrán ser consultados por cualquier persona. La modificación de categorías sólo puede realizarla el encargado del catálogo y el administrador.	
Derechos de acceso:	La actualización será previa autorización con nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
Cat_id	Identificador de la categoría	Int
Cat_categoria	Nombre de la categoría	String

Entidad:	Subcategoría	
Descripción:	Almacenar toda la información relativa a las subcategorías de los productos.	
Número de ejemplares:	38	
Crecimiento previsto:	5 subcategorías/año	
Confidencialidad:	Los datos podrán ser consultados por cualquier persona. La modificación de subcategorías sólo puede realizarla el encargado del catálogo y el administrador.	
Derechos de acceso:	La actualización será previa autorización con nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
Subcat_id	Identificador de la subcategoría	Int
Cat_id	Identificador de la categoría	String
Subcat_subcategoría	Nombre de la categoría	

Entidad:	Marca	
Descripción:	Almacenar toda la información relativa a las marcas de los productos.	
Número de ejemplares:	15	
Crecimiento previsto:	5 marcas/año	
Confidencialidad:	Los datos podrán ser consultados por cualquier persona. La modificación de marcas sólo puede realizarla el encargado del catálogo y el administrador.	
Derechos de acceso:	La actualización será previa autorización con nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
Marca_id	Identificador de la marca.	Int
Marca_nombre	Nombre de la marca.	String
Marca_rutaimg	Ruta de la imagen correspondiente al logotipo.	String

Entidad:	Promoción	
Descripción:	Almacenar las promociones aplicables a los productos en un determinado periodo de tiempo.	
Número de ejemplares:	100	
Crecimiento previsto:	10 promociones/mes	
Confidencialidad:	Los datos podrán ser consultados por cualquier persona. La modificación de promociones sólo puede realizarla el encargado del catálogo y el administrador.	
Derechos de acceso:	La actualización será previa autorización con nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
Promo_id	Identificador de la promoción.	Int
Prod_id	Identificador del producto en promoción.	String
Promo_desc	Descuento aplicado al precio del producto.	String
Promo_fechaIn	Fecha en que inicia la promoción.	String
Promo_fechaFin	Fecha en que termina la promoción.	String

Entidad:	Almacén	
Descripción:	Almacenar los datos referentes a cada producto en almacén.	
Número de ejemplares:	228	
Crecimiento previsto:	5 registros nuevos/mes	
Confidencialidad:	Los datos podrán ser consultados por el encargado del almacén y el administrador. La modificación sólo puede realizarla el encargado del almacén y el administrador.	
Derechos de acceso:	La actualización será previa autorización con nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
Alm_id	Identificador de el producto en el almacén	Int
Prod_id	Identificador del producto en almacén.	Int
Alm_exist	Existencias de ese productos.	Int
Alm_fechaexist	Fecha en que se hizo el último conteo de existencias.	Date
Alm_precioent	Último precio de entrada registrado.	Float
Alm_preciosalida	Último precio de salida registrado.	Float

Entidad:	Inventario	
Descripción:	Almacenar la información del inventario de productos.	
Número de ejemplares:	-	
Crecimiento previsto:	-	
Confidencialidad:	Los datos podrán ser consultados por el encargado del almacén y el administrador. La modificación sólo puede realizarla el encargado del almacén y el administrador.	
Derechos de acceso:	La actualización será previa autorización con nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
Prod_id	Identificador del producto en el inventario.	Int
Inv_numserie	Número de serie del producto, son únicos.	String
Inv_fechaent	Fecha de entrada al almacén.	Date
Inv_fechaasal	Fecha de salida del almacén.	Date
Inv_precioent	Precio de entrada.	Float
Inv_preciosal	Precio de salida.	Float

Entidad:	Equipo	
Descripción:	Almacenar la información de los equipos de cómputo ensamblados por Sumitel.	
Número de ejemplares:	-	
Crecimiento previsto:	-	
Confidencialidad:	Los datos podrán ser consultados por el encargado del almacén y el administrador. La modificación sólo puede realizarla el encargado del almacén y el administrador.	
Derechos de acceso:	La actualización será previa autorización con nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
Eq_id	Identificador del equipo.	Int
Prod_id	Identificador del componente.	Int
Inv_serie	Número de serie del componente.	String

Entidad:	Orden_Compra	
Descripción:	Almacena los datos de las órdenes de compra.	
Número de ejemplares:	-	
Crecimiento previsto:	-	
Confidencialidad:	Los datos podrán ser consultados por el encargado del almacén y el administrador. La modificación sólo puede realizarla el encargado del almacén y el administrador.	
Derechos de acceso:	La actualización será previa autorización con nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
Ord_id	Identificador de la orden de compra.	Int
Ord_status	Estatus de la orden de compra.	Int
Ord_fecha	Fecha de entrada la orden.	Date
Pv_id	Identificador del proveedor de esa orden de compra.	Int

Entidad:	Detalle_Orden_Compra	
Descripción:	Almacena las partidas de las órdenes de compra.	
Número de ejemplares:	-	
Crecimiento previsto:	-	
Confidencialidad:	Los datos podrán ser consultados por el encargado del almacén y el administrador. La modificación sólo puede realizarla el encargado del almacén y el administrador.	
Derechos de acceso:	La actualización será previa autorización con nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
Ord_id	Identificador de la orden de compra que se esta detallando	Int
Prod_id	Identificador del producto que se va a comprar	Int
Prod_cant	Cantidad a comprar	Int
Do_costo	Costo	Float
Do_desc	Descuento aplicado a este producto	Float
Do_iva	IVA de este producto	Float
Do_tot	Importe total	Float()

Entidad:	Devolucion	
Descripción:	Almacena los datos de las devoluciones de productos.	
Número de ejemplares:	-	
Crecimiento previsto:	-	
Confidencialidad:	Los datos podrán ser consultados por el encargado del almacén y el administrador. La modificación sólo puede realizarla el encargado del almacén y el administrador.	
Derechos de acceso:	La actualización será previa autorización con nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
Ord_id	Identificador de la orden de compra con la que sumitel adquirió el producto devuelto.	Int
Dev_id	Identificador de la devolución.	Int
Ped_id	Identificador del pedido con el que el cliente compró este producto.	Int
Dev_fecha	Fecha en que se esta registrando la devolución.	Date
Prod_id	Identificador del producto devuelto.	Int
Alm_numser	Número de serie del producto.	String

Entidad:	Proveedor	
Descripción:	Almacena la información de los proveedores de productos.	
Número de ejemplares:	10	
Crecimiento previsto:	5 proveedores/año	
Confidencialidad:	Los datos podrán ser consultados por los vendedores, el encargado del almacén y el administrador. La modificación sólo puede realizarla el encargado del almacén y el administrador.	
Derechos de acceso:	La actualización será previa autorización con nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
Pv_id	Identificador del proveedor	Int
Pv_contacto	Identificador del contacto.	
Pv_nom	Nombre del proveedor	String
Pv_dir	Dirección del proveedor	Int
Pv_tel	Teléfono del proveedor	String
Pv_rfc	RFC del proveedor	String

Entidad:	ProductoProveedor	
Descripción:	Relaciona los productos con uno o más proveedores.	
Número de ejemplares:	-	
Crecimiento previsto:	-	
Confidencialidad:	Los datos podrán ser consultados por el encargado del almacén y el administrador. La modificación sólo puede realizarla el encargado del almacén y el administrador.	
Derechos de acceso:	La actualización será previa autorización con nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
Prod_id	Identificador del producto	Int
Prov_id	Identificador del proveedor	Int

Entidad:	Persona	
Descripción:	Almacena la información que define a una persona del sistema.	
Número de ejemplares:	1000	
Crecimiento previsto:	50 clientes/mes	
Confidencialidad:	Los datos podrán ser consultados por el sistema únicamente.	
Derechos de acceso:	La actualización será previa autorización con nombre de usuario y contraseña solo cuando la tabla derivada sea actualizada.	
Atributo	Descripción	Tipo
per_id	Identificador de la persona.	Int
per_nombre	Nombre.	String
per_ap_paterno	Apellido paterno.	String
per_ap_materno	Apellido materno.	String
per_fecha_nacimiento	Fecha de nacimiento.	Date
per_estado_civil	Estado civil (Soltero-1, Casado-2, Divorciado-3, Viudo-4, Otro-5).	Int
per_correo1	Dirección de correo electrónico.	String
per_correo2	Dirección de correo electrónico alterna.	String
per_tel_particular	Teléfono particular.	String
per_tel_oficina	Teléfono de la oficina.	String
per_curp	Clave Única de Registro de Población.	String

Entidad:	Cliente	
Descripción:	Almacena la información que define a un cliente.	
Número de ejemplares:	800	
Crecimiento previsto:	80 clientes/mes	
Confidencialidad:	Los datos podrán ser consultados por el cliente mismo, los vendedores y el administrador. La modificación sólo puede realizarla un vendedor, el mismo cliente o el administrador.	
Derechos de acceso:	La actualización será previa autorización con nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
Cli_id	Identificador del cliente	Int
Dom_fiscal	Identificador del domicilio fiscal.	Int
Per_id	Identificador de la persona.	Int
Grp_id	Grupo de cliente.	Int
Cli_RFC	Registro Federal de Causantes.	String
Cli_persona_fiscal	Tipo de persona fiscal (Moral-1, Física-2)	Int

Entidad:	Pago	
Descripción:	Almacena una bitácora de los pagos realizados por un cliente.	
Número de ejemplares:	800	
Crecimiento previsto:	-	
Confidencialidad:	Los datos podrán ser consultados por el cliente mismo, los vendedores y el administrador. La modificación sólo puede realizarla un vendedor, el mismo cliente o el administrador.	
Derechos de acceso:	La actualización será previa autorización con nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
Ped_id	Identificador del pedido	Int
Pago_fecha	Fecha en que se realizó el pago. dd-mm-aaaa hh:mm:ss.	Int
Pago_importe	Importe del pago.	Float

Entidad:	GrupoCliente	
Descripción:	Almacena el catálogo de grupos de clientes.	
Número de ejemplares:	800	
Crecimiento previsto:	-	
Confidencialidad:	Los datos podrán ser consultados por el cliente mismo, los vendedores y el administrador. La modificación sólo puede realizarla un vendedor, el mismo cliente o el administrador.	
Derechos de acceso:	La actualización será previa autorización con nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
Grp_id	Identificador del grupo.	Int
Grp_desc	Descuento que se aplicará al importe de los pedidos de este grupo de clientes.	Float
Grp_grupo	Nombre del grupo.	String

Entidad:	Mensajería	
Descripción:	Almacena el catálogo de empresas de mensajería.	
Número de ejemplares:	2.	
Crecimiento previsto:	1 empresas/año.	
Confidencialidad:	Los datos podrán ser consultados por los vendedores, el encargado del almacén y el administrador. La modificación sólo puede realizarla un vendedor, el encargado del almacén o el administrador.	
Derechos de acceso:	La actualización será previa autorización con nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
Mens_id	Identificador de la empresa.	Int
Mens_nombre	Nombre de la empresa.	String
Mens_contacto	Identificador de la persona de contacto.	Int
Mens_tel	Teléfono.	String
Mens_correo	Dirección de correo electrónico.	String

Entidad:	UsuariosSistema	
Descripción:	Almacena el catálogo usuarios del sistema.	
Número de ejemplares:	15.	
Crecimiento previsto:	5 usuarios/año.	
Confidencialidad:	Los datos podrán ser consultados por el administrador y el mismo usuario. La modificación sólo puede realizarla un usuario, o el administrador.	
Derechos de acceso:	La actualización será previa autorización con nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
Usr_id	Identificador del usuario.	Int
Per_id Integer	Identificador de la persona	Int
Usr_area	Área dentro de la tienda en la que se desarrolla el usuario.	String

Entidad:	PermisoUsuario	
Descripción:	Contiene los datos de acceso al sistema.	
Número de ejemplares:	15.	
Crecimiento previsto:	5 usuarios/año.	
Confidencialidad:	Los datos podrán ser consultados por el administrador y el mismo usuario. La modificación sólo puede realizarla un usuario (restringida), o el administrador.	
Derechos de acceso:	La actualización será previa autorización con nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
Usr_id	Identificador del usuario.	Int
Perm_nombre_usuario	Nombre de usuario para acceso al sistema.	String
Perm_contraseña	Contraseña para acceso al sistema.	String
Perm_tipo_usuario	Tipo de usuario.	String
Perm_hora_ini	Hora de entrada al sistema.	Date
Perm_hora_fin	Hora de cierre del sistema para este usuario.	Date
Perm_estatus	Estatus del usuario (Activo-1, Inactivo-0).	Int

Entidad:	Domicilio	
Descripción:	Almacena los domicilios de los clientes.	
Número de ejemplares:	-	
Crecimiento previsto:	-	
Confidencialidad:	Los datos podrán ser consultados por el vendedor, encargado de almacén, cliente y el. La modificación sólo puede realizarla un cliente, un vendedor o el administrador.	
Derechos de acceso:	La actualización será previa autorización con nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
Dom_id	Identificador del domicilio.	Int
Dom_calle	Calle	String
Dom_num_ext	Número exterior del domicilio.	String
Dom_num_int	Número interior del domicilio.	String
Dom_col	Colonia.	String
Dom_mun	Municipio o delegación.	String
Dom_estado	Estado.	String
Dom_pais	País.	String
Dom_cp	Código postal.	String
Dom_tel	Teléfono.	String

Entidad:	FormaPago	
Descripción:	Almacena las opciones de pago para cada cliente.	
Número de ejemplares:	-	
Crecimiento previsto:	-	
Confidencialidad:	Los datos podrán ser consultados por el vendedor, cliente y el. La modificación sólo puede realizarla un cliente, un vendedor o el administrador.	
Derechos de acceso:	La actualización será previa autorización con nombre de usuario y contraseña.	
Atributo	Descripción	Tipo
Fpago_id	Identificador del registro.	Int
Cli_id	Identificador del cliente.	String
Fpago_forma	Forma de pago.	String
Fpago_tipo_tarjeta	Tipo tarjeta.	String
Fpago_titular	Titular de la cuenta.	String
Fpago_banco	Banco.	String
Fpago_num_tarjeta	Número de tarjeta.	String
Fpago_verificador	Dígito verificador.	String

2.5 Modelado

El modelado es la etapa siguiente del ciclo de vida del sistema. Puede llevarse a cabo basado en los enfoques tradicionales o un enfoque orientado a objetos.

Este último enfoque permite que muchas de las características deseables en un sistema informático puedan cumplirse. El modelado orientado a objetos ofrece resolver problemas en función de la representación de objetos con características que lo hacen único dentro de un entorno y sus relaciones con otros objetos. Se crean objetos independientes con funciones específicas que podemos llamar componentes, cuya principal ventaja es la reutilización en otros sistemas.

Dos conceptos importantes en el modelado de objetos son encapsulamiento, polimorfismo y herencia.

El encapsulamiento se refiere a la capacidad que tendrá un objeto para ocultar sus atributos y su comportamiento, esto es, todos los demás objetos solo conocerán ciertos atributos del objeto en cuestión y sólo podrán hacerlo a través de un método.

Polimorfismo es la habilidad que tiene un objeto para responder de manera diferente a un estímulo en particular. En otras palabras, un método de una clase puede regresar diferentes resultados dependiendo del mensaje o parámetro que se le envíe.

Una de las ventajas del modelado orientado a objetos es que podemos especializar las clases tanto como queramos. Se define una clase base, lo más abstracta posible, después, se definen otras clase derivadas de ésta pero con ciertas características adicionales que lo hacen diferente a otras derivadas de la misma. Esto se conoce como herencia.

2.5.1. Pasos para el modelado orientado a objetos

El modelado orientado a objetos inicia con la identificación de los objetos, sus particularidades, interacción y comportamiento dentro del entorno.

Una vez identificado lo anterior podemos definir una clase. Una clase define las características de un objeto, una instancia es un ejemplar de una clase.

Los objetos pueden tener diferentes tipos de relaciones o asociaciones, dependiendo de la cardinalidad y dependencia.

El siguiente paso es definir el modelo de requerimientos, conformado por:

- Modelo de casos de uso.

Extrae el conocimiento funcional básico del problema y establece la estructura base del sistema. La funcionalidad está dada por transacciones realmente significativas, a cada una de estas transacciones se le denomina caso de uso. El nombre del caso de uso debe describir en unas cuantas palabras la finalidad de la transacción.

Todos los casos de uso son ejecutados por un actor al menos.

Un actor no necesariamente es una persona y no forma parte del sistema, sólo interactúa con él.

Es necesario identificar los flujos alternos en los que pueda desembocar una transacción y no olvidar tomar todas las alternativas posibles.

Concretando, un caso de uso nace del ¿quién y cómo ejecutará una transacción?; debe tener un *nombre, una secuencia de pasos y sus flujos alternos o de excepción*.

- Modelo de interfaz.

Establece el vínculo visual entre el desarrollador y el usuario a través de la interacción que el sistema tenga con el entorno.

Esencialmente, la interacción con el usuario se logra a través de pantallas, reportes y llamadas a subsistemas.

Las pantallas son estáticas en este punto del ciclo de desarrollo, sólo deben dar una idea al usuario de lo que verá como resultado final.

Los reportes se definirán en base a los datos obtenidos en el diagrama de entidad-relación, tomando sólo aquellos que sean significativos para el usuario. También deben considerarse todos aquellos reportes que se soliciten esporádicamente.

- Modelo del dominio del problema. Se definen los principales objetos que forman el sistema y sus relaciones entre sí.

Estas relaciones tienen dirección, nomenclatura y cardinalidad.

2.5.2. UML

UML es una herramienta para el modelado de objetos y básicamente son diagramas que permiten entender de manera más clara el comportamiento del sistema y la interacción entre sus componentes, antes de comenzar a programar.

Existen diagramas para cada una de las fases del modelado.

Los diagramas que se cubrieron durante este módulo fueron:

- Diagrama de casos de uso
- Diagramas de secuencia
- Diagramas de clases
- Diagramas de colaboración
- Diagramas de estado
- Diagramas de componentes
- Diagramas de despliegue

Durante el módulo se realizaron ejercicios de cada uno de los diagramas, tanto para el problema de ejemplo (venta de boletos) como para algunos de los proyectos propuestos por los asistentes al diplomado.

El cuadro 2.1 describe de manera más amplia los diagramas que conforman el UML.

Área	Vista	Diagramas	Conceptos Principales
Estructural	Vista Estática	Diagrama de Clases	Clase, asociación, generalización, dependencia, realización, interfaz.
	Vista de Casos de Uso	Diagramas de Casos de Uso	Caso de Uso, Actor, asociación, extensión, generalización.
	Vista de Implementación	Diagramas de Componentes	Componente, interfaz, dependencia, realización.
	Vista de Despliegue	Diagramas de Despliegue	Nodo, componente, dependencia, localización.
Dinámica	Vista de Estados de máquina	Diagramas de Estados	Estado, evento, transición, acción.
	Vista de actividad	Diagramas de Actividad	Estado, actividad, transición, determinación, división, unión.
	Vista de interacción	Diagramas de Secuencia	Interacción, objeto, mensaje, activación.
		Diagramas de Colaboración	Colaboración, interacción, rol de colaboración, mensaje.
Administración o Gestión de modelo	Vista de Gestión de modelo	Diagramas de Clases	Paquete, subsistema, modelo.
Extensión de UML	Todas	Todos	Restricción, estereotipo, valores, etiquetados.

Cuadro 2.1 Diagramas UML

2.5.3 Listado de los casos de uso

El siguiente listado muestra los casos de uso identificados para el caso práctico de Sumitel.

Nombre
1. Buscar un producto por categoría.
2. Buscar un producto por palabra.
3. Buscar un producto por marca.
4. Buscar un producto por búsqueda avanzada.
5. Consultar detalles del producto.
6. Registrar nuevo cliente en sitio.
7. Agregar productos a carrito de compras.
8. Confirmar compra.
9. Efectuar pago.
10. Iniciar sesión en sitio e-commerce.
11. Buscar un pedido en sitio e-commerce.
12. Buscar un pedido en sitio de administración.
13. Buscar pedidos de un cliente en el sitio de administración.
14. Consultar detalles de un pedido.
15. Rastrear el envío de un pedido en el sitio e-commerce.
16. Rastrear el envío de un pedido en el sitio de administración.
17. Devolver un pedido.
18. Actualizar datos del cliente.
19. Verificar pedidos pendientes.
20. Solicitar una salida al almacén.
21. Registrar recepción del pedido en sitio de administración.
22. Enviar solicitud de Cancelación de un pedido.
23. Actualizar el estado de un pedido.
24. Mostrar promociones.
25. Recuperar contraseña.
26. Recibir solicitud de salida de almacén.
27. Actualizar entradas al almacén.
28. Entregar pedido al porteador (Empresa de Mensajería).
29. Actualiza entradas/salidas almacén
30. Dar de alta usuarios del sistema
31. Dar de baja usuarios del sistema
32. Cambios a las cuentas de usuarios.
33. Crear categoría de productos.
34. Asignar permisos a las cuentas de usuarios.

2.5.4 Listado de Casos de Uso y Actores

Cada caso de uso se ha asociado con los actores del sistema, quedando de la siguiente manera.

Actor Primario	Casos de Uso
Cliente Visitante	Buscar productos, registrarse en sitio, consultar detalles de un producto, agregar productos a carrito de compras.
Cliente Comprador	Buscar productos, Consultar detalles del producto, Agregar productos a carrito de compras, Confirmar compra, Iniciar sesión en sitio e-commerce, Efectuar pago, Buscar un pedido en sitio e-commerce, Consultar detalles de un pedido, Rastrear el envío de un pedido en el sitio e-commerce, Devolver un pedido, Actualizar datos del cliente, Verificar pedidos pendientes, Enviar solicitud de Cancelación de un pedido, Recuperar contraseña.
Vendedor	Buscar productos, Consultar detalles del producto, Agregar productos a carrito de compras, Confirmar compra, Efectuar pago, Buscar un pedido en sitio de administración, Consultar detalles de un pedido, Buscar pedidos de un cliente en el sitio de administración, Actualizar datos del cliente, Verificar pedidos pendientes, Solicitar una salida al almacén, Registrar recepción del pedido en sitio de administración, Actualizar el estado de un pedido, Cancelar un pedido.
Almacenista	Recibir solicitud de salida de almacén, Actualizar entradas al almacén, Actualizar salidas del almacén, Entregar pedido al porteador (Empresa de Mensajería), Asignar número de pedido.
Responsable del Catálogo de Productos (RCP)	Dar de alta catálogo de productos, dar de baja productos en el catálogo, Actualizar datos en el catálogo de productos.
Administrador	Dar de alta a los usuarios (Sistema), Dar de baja a los usuarios, Actualizar status de usuarios, Crear categorías de productos, Dar de baja categorías de productos, Asignar permisos a los usuarios, Generar reportes varios en el sistema, Dar de baja clientes, Actualizar catálogo de promociones.

2.5.5 Definición de los casos de uso.

ID del Caso de Uso:	1		
Nombre del Caso de Uso:	Buscar un producto por categoría.		
Creado Por:	Svicente	Fecha de Creación:	05/11/2004

Actores:	Cliente Visitante, Cliente Comprador, Vendedor, RCP, Almacenista, Administrador
Descripción:	El actor entra al sitio Web de la tienda. Busca en el catálogo de productos aquellos que sean de su interés. Tiene las siguientes opciones: <i>buscar por categorías, por palabra, por marca y por búsqueda combinada (avanzada).</i>
Precondiciones:	<ul style="list-style-type: none"> ▪ Haber entrado al sitio Web. ▪ Login no necesario. ▪ Debe haber al menos una categoría capturada. ▪ Puede haber subcategorías capturadas. ▪ Mostrar un listado de las categorías disponibles que tengan subcategorías con productos o productos.
Post-condiciones:	Listado de productos que pertenecen a una categoría o subcategoría.
Flujo Normal:	<ol style="list-style-type: none"> 1. El actor entra al sitio Web. 2. Se muestra un menú con las categorías con productos existentes. 3. El actor elige una categoría. 4. Se muestran los productos de esta categoría.
Flujos Alternos:	<ol style="list-style-type: none"> 3. Si la categoría tiene subcategorías y éstas tienen productos asignados, se muestran en un listado. 4. El actor puede elige una subcategoría. 5. Se muestran los productos de esta subcategoría. <p>5. Si no hay productos capturados en la categoría o subcategoría, se muestra un mensaje al usuario y se le invita a elegir otra categoría.</p>
Excepciones:	<p>No hay productos asignados a la categoría elegida.</p> <p>No hay categorías capturadas.</p>
Observaciones:	Las subcategorías deben tener al menos un producto. En caso contrario no deben ser listadas en el menú.

ID del Caso de Uso:	2		
Nombre del Caso de Uso:	Buscar un producto por palabra.		
Creado Por:	Svicente	Fecha de Creación:	05/11/2004

Actores:	Cliente Visitante, Cliente Comprador, Vendedor, Almacenista, RCP, Administrador
Descripción:	El actor entra al sitio Web de la tienda y busca un producto que contenga una palabra clave.
Precondiciones:	<ul style="list-style-type: none"> ▪ Haber entrado al sitio Web. ▪ Login no necesario. ▪ Debe haber al menos un producto capturado. ▪ Mostrar un formulario de búsqueda con un cuadro de texto y un botón de envío.
Poscondiciones:	Listado de productos cuya clave, descripción, marca o modelo coincidan con la palabra clave.
Flujo Normal:	<ol style="list-style-type: none"> 1. El actor entra al sitio Web. Teclea una palabra en el formulario de búsqueda por palabras. 2. El sistema busca aquellos productos en cuya clave, descripción, marca o modelo, aparezca alguna ocurrencia de la palabra clave. 3. Se muestran los productos encontrados.
Flujos Alternos:	<ol style="list-style-type: none"> 2. Si el cliente no proporciona una palabra, mandar mensaje de error. 3. Si no hay coincidencias, mostrar mensaje. Mostrarle productos aleatorios.
Excepciones:	
Observaciones:	

ID del Caso de Uso:	3		
Nombre del Caso de Uso:	Buscar un producto por marca.		
Creado Por:	Svicente	Fecha de Creación:	05/11/2004

Actores:	Cliente Visitante, Cliente Comprador
Descripción:	El actor entra al sitio Web de la tienda. Busca en el catálogo de productos aquellos que sean de su interés. Tiene las siguientes opciones: buscar por categorías, por palabra, por marca y por búsqueda combinada (avanzada).
Precondiciones:	<ul style="list-style-type: none"> ▪ Entrar al sitio Web mediante un navegador Web. ▪ Mostrar un formulario de búsqueda con una lista de las marcas disponibles.
Poscondiciones:	Listado de productos cuya marca coinciden con la marca seleccionada en el formulario de búsqueda.
Flujo Normal:	<ol style="list-style-type: none"> 1. El cliente (Visitante o Comprador) entran al sitio Web. Selecciona una marca en el formulario de búsqueda por marca. 2. Buscar aquellos productos cuya marca asignada coincida con la marca seleccionada. 3. Se muestran los productos encontrados.
Flujos Alternos:	<p>Si el cliente no selecciona una marca, mandar mensaje de error.</p> <p>Si no hay coincidencias, mostrar mensaje. Mostrarle productos aleatorios.</p>
Excepciones:	
Observaciones:	

ID del Caso de Uso:	4		
Nombre del Caso de Uso:	Buscar un producto por búsqueda avanzada.		
Creado Por:	Svicente	Fecha de Creación:	05/11/2004

Actores:	Cliente Visitante, Cliente Comprador
Descripción:	El actor entra al sitio Web de la tienda. Busca en el catálogo de productos aquellos que sean de su interés. Tiene las siguientes opciones: buscar por categorías, por palabra, por marca y por búsqueda combinada (avanzada).
Precondiciones:	Entrar al sitio Web mediante un navegador Web. Mostrar un formulario de búsqueda con los siguientes campos: <ul style="list-style-type: none"> - Lista de Categorías - Lista de Subcategorías - Lista Marcas - Lista rango de precios
Poscondiciones:	Listado de productos que coincidan con los criterios introducidos en el formulario de búsqueda.
Flujo Normal:	<ol style="list-style-type: none"> 1. El cliente (Visitante o Comprador) entran al sitio Web. Selecciona las opciones que desea en el formulario de búsqueda avanzada. 2. Se combinan los campos de búsqueda seleccionados y se buscan aquellos productos que cumplan con esos criterios. 3. Se muestran los productos encontrados.
Flujos Alternos:	<ol style="list-style-type: none"> 1. Si el cliente no selecciona por lo menos un criterio, mandar mensaje de error y no procesar la búsqueda. 2. Si no hay coincidencias, mostrar mensaje. Mostrarle productos aleatorios.
Excepciones:	
Observaciones:	

ID del Caso de Uso:	5		
Nombre del Caso de Uso:	Consultar detalles de un producto.		
Creado Por:	Svicente	Fecha de Creación:	05/11/2004

Actores:	Cliente Visitante, Cliente Comprador
Descripción:	Una vez que el cliente encontró los productos de su interés, podrá consultar la ficha que contiene los detalles de cada uno.
Precondiciones:	Haber realizado una búsqueda (por categoría, por marca o avanzada).
Poscondiciones:	Ficha con detalles del producto: imagen, descripción completa, precio.
Flujo Normal:	<ol style="list-style-type: none"> 1. El cliente (Visitante o Comprador) da clic a la imagen o a la descripción corta del producto. 2. Se busca el producto en la base de datos. 3. Se despliegan los datos en una página. 4. Si el hay stock de este producto se muestra el botón 'agregar a mi carrito de compras'.
Flujos Alternos:	<ol style="list-style-type: none"> 4. Si el no hay stock, no se muestra el botón 'agregar a mi carrito de compras'.
Excepciones:	Si el id del producto es erróneo por alguna razón y en consecuencia el producto no se encuentra en la base de datos, enviar un mensaje al cliente notificándole que no existe ese producto y mostrar un listado de productos aleatorios de la categoría seleccionada.
Observaciones:	

ID del Caso de Uso:	6		
Nombre del Caso de Uso:	Registrar nuevo cliente en sitio.		
Creado Por:	Svicente	Fecha de Creación:	05/11/2004

Actores:	Cliente Visitante
Descripción:	Un cliente visitante se da de alta para que sus datos queden almacenados en la base de datos del sistema ya sea para recibir promociones, o bien, para no tener que capturar sus datos cada vez que realice una compra.
Precondiciones:	Haber entrado al sitio. Elegir la opción "Registro de Clientes"
Poscondiciones:	Cliente registrado y con una sesión abierta.
Flujo Normal:	<ol style="list-style-type: none"> 1. El cliente visitante elige la opción "Registro de Clientes". 2. Se despliega un formulario con tres secciones: <ol style="list-style-type: none"> a. Datos de la cuenta: nombre de usuario y contraseña. b. Datos personales: información personal y domicilio particular. c. Datos de envío: domicilio de envío por default, una vez registrado podrá agregar más de un domicilio de envío. 3. El cliente envía la información. 4. El sistema valida la información. El nombre de usuario no debe existir con anterioridad así como el correo electrónico. Se debe validar que la contraseña y nombre de usuario sean válidos. 5. Los datos del cliente se guardan en la base de datos. 6. El sistema envía un correo de bienvenida a la dirección de correo electrónico proporcionada, este debe contener el nombre de usuario y la contraseña del cliente registrado. 7. Se inicia una sesión automáticamente. 8. Se redirecciona al cliente a la sección 'Mi Cuenta'.
Flujos Alternos:	<ol style="list-style-type: none"> 4.1 Si los datos no son válidos, se muestra nuevamente la pantalla con los datos proporcionados, menos los datos errados y la contraseña. Se muestra un mensaje de error notificándole los datos que son erróneos y alguna sugerencia. 4.2 Si el nombre de usuario ya existe, se muestra nuevamente la pantalla con los datos proporcionados y se sugieren nombres de usuario válidos y que no existan actualmente en la base de datos. 4.3 Si el cliente se está registrando porque va a concretar una compra, al terminar el registro se le redirecciona a la sección 'Confirmar compra' en vez de 'Mi cuenta'.
Excepciones:	
Observaciones:	

ID del Caso de Uso:	7		
Nombre del Caso de Uso:	Agregar productos a carrito de compras.		
Creado Por:	Svicente	-	Fecha de Creación: 05/11/2004

Actores:	Cliente Visitante, Cliente Comprador
Descripción:	Para realizar una compra, un cliente debe agregar productos a su carrito, ya sea desde el listado de la búsqueda, promociones o desde la ficha de detalles.
Precondiciones:	<ul style="list-style-type: none"> ▪ Haber realizado una búsqueda. ▪ Consultar la ficha de detalles. ▪ Seleccionar producto desde algún listado de promociones.
Poscondiciones:	Carrito de compras con al menos un producto.
Flujo Normal:	<ol style="list-style-type: none"> 1. El cliente realiza una búsqueda de productos. 2. Se muestra el listado de productos. 3. Hace clic en el producto que le interesa. 4. Se despliega la ficha de detalles. 5. El cliente hace clic en el botón 'agregar a mi carrito de compras'. 6. El id del producto seleccionado es agregado al contenedor del carrito. 7. Por defecto la cantidad del producto agregado es 1.
Flujos Alternos:	<ol style="list-style-type: none"> 1. El cliente realiza una búsqueda de productos. 2. Se muestra el listado de productos. 3. El cliente hace clic en el botón 'agregar a mi carrito de compras' correspondiente al producto que le interesa. 4. El id del producto seleccionado es agregado al contenedor del carrito. 5. Por defecto la cantidad del producto agregado es 1. <ol style="list-style-type: none"> 1. Se despliega una oferta de manera aleatoria. 2. El cliente da clic en el botón de detalles. 3. El cliente hace clic en el botón 'agregar a mi carrito de compras'. 4. El id del producto seleccionado es agregado al contenedor del carrito. 5. Por defecto la cantidad del producto agregado es 1.
Excepciones:	Si el producto no tiene existencias en la bodega, no se despliega el botón 'agregar a mi carrito de compras'.
Observaciones:	

ID del Caso de Uso:	8		
Nombre del Caso de Uso:	Confirmar compra.		
Creado Por:	Svicente	Fecha de Creación:	07/11/2004

Actores:	Cliente Visitante, Cliente Comprador
Descripción:	Una vez que el cliente tiene en su carrito de compras todos los productos que desea adquirir, debe confirmar su compra.
Precondiciones:	<ul style="list-style-type: none"> ▪ El carrito de compras debe tener al menos un producto. ▪ El cliente debe iniciar una sesión. ▪ El cliente debe estar registrado.
Poscondiciones:	Pedido autorizado por el cliente y listo para procesar el pago del mismo.
Flujo Normal:	<ol style="list-style-type: none"> 1. Una vez que el cliente ha agregado sus productos al carrito de compras, hace clic en el botón 'procesar compra'. 2. Se pregunta al cliente que seleccione la dirección a la que desea que se le envíe su pedido. 3. Después se despliega una pantalla con sus datos personales, la dirección de envío que seleccionó y el detalle de su pedido (productos, cantidad, precio unitario, importe) y los subtotales. 4. En la misma pantalla se muestra el formulario de forma de pago. En éste, el cliente debe seleccionar el tipo de tarjeta de crédito, teclear el número, la fecha de expiración, el código verificador y el nombre que aparece en su tarjeta de crédito. 5. Para terminar la compra, debe hacer clic en el botón 'terminar'. 6. Los datos se envían a través de una conexión segura.
Flujos Alternos:	<ol style="list-style-type: none"> 1. Una vez que el cliente ha agregado sus productos al carrito de compras, hace clic en el botón 'procesar compra'. 2. Se despliega la pantalla de inicio de sesión. 3. El cliente debe ingresar su nombre de usuario y su contraseña. 4. Una vez autorizado, se despliega una pantalla con sus datos personales, la dirección de envío que seleccionó y el detalle de su pedido (productos, cantidad, precio unitario, importe) y los subtotales. 5. En la misma pantalla se muestra el formulario de forma de pago. En éste, el cliente debe seleccionar el tipo de tarjeta de crédito, teclear el número, la fecha de expiración, el código verificador y el nombre que aparece en su tarjeta de crédito. 6. Para terminar la compra, debe hacer clic en el botón 'terminar'. 7. Los datos se envían a través de una conexión segura. <ol style="list-style-type: none"> 1. Una vez que el cliente ha agregado sus productos al carrito de compras, hace clic en el botón 'procesar compra'. 2. Se despliega la pantalla de inicio de sesión. 3. El cliente no se ha registrado aún, por lo tanto selecciona la opción 'registro de clientes'. 4. El cliente se registra en el sitio y se inicia su sesión. 5. Se despliega una pantalla con sus datos personales, la dirección de envío que seleccionó y el detalle de su pedido (productos, cantidad, precio unitario, importe) y los subtotales. 6. En la misma pantalla se muestra el formulario de forma de pago. En éste, el cliente debe seleccionar el tipo de tarjeta de crédito, teclear el número, la fecha de expiración, el código verificador y el nombre que aparece en su tarjeta de crédito. 7. Para terminar la compra, debe hacer clic en el botón 'terminar'. 8. Los datos se envían a través de una conexión segura.
Excepciones:	<p>Si el cliente no está registrado, darle la opción de registrarse.</p> <p>Si el cliente extravió su contraseña, darle la opción de recuperar ambos proporcionando su correo electrónico y su nombre de usuario.</p>
Observaciones:	

ID del Caso de Uso:	9		
Nombre del Caso de Uso:	Efectuar pago.		
Creado Por:	Svicente	Fecha de Creación:	07/11/2004

Actores:	Cliente Visitante, Cliente Comprador
Descripción:	El cliente proporciona la información necesaria para pagar su pedido. Estos datos son autorizados por un tercero y dependiendo de la respuesta se procederá a crear el pedido o pedir otra forma de pago.
Precondiciones:	<ul style="list-style-type: none"> ▪ El carrito de compras debe tener al menos un producto. ▪ El cliente debe haber iniciado una sesión. ▪ El cliente debe estar registrado. ▪ El cliente debe haber seleccionado la opción 'procesar pedido'.
Poscondiciones:	Pago del pedido autorizado o rechazado. Pedido creado o pedido cancelado.
Flujo Normal:	<ol style="list-style-type: none"> 1. Después de seleccionar la dirección a la que desea que se le envíe su pedido, se despliega una pantalla con sus datos personales, la dirección de envío que seleccionó y el detalle de su pedido (productos, cantidad, precio unitario, importe) y los subtotales. 2. En la misma pantalla se muestra el formulario de forma de pago. En éste, el cliente debe seleccionar el tipo de tarjeta de crédito, teclear el número, la fecha de expiración, el código verificador y el nombre que aparece en su tarjeta de crédito. 3. Para terminar la compra, debe hacer clic en el botón 'terminar'. 4. Los datos se envían a través de una conexión segura a una Terminal virtual. 5. La aplicación para manejo de transacciones con tarjetas de crédito se encarga de autorizar el pago. 6. Si el pago se realizó satisfactoriamente se crea un nuevo pedido. 7. Se envía un correo electrónico al cliente notificándole que su pago fue aceptado y que su pedido está siendo surtido. 8. Se envía un correo al departamento de ventas para asignar un vendedor al nuevo pedido.
Flujos Alternos:	<ol style="list-style-type: none"> 1. Después de seleccionar la dirección a la que desea que se le envíe su pedido, se despliega una pantalla con sus datos personales, la dirección de envío que seleccionó y el detalle de su pedido (productos, cantidad, precio unitario, importe) y los subtotales. 2. En la misma pantalla se muestra el formulario de forma de pago. En éste, el cliente debe seleccionar el tipo de tarjeta de crédito, teclear el número, la fecha de expiración, el código verificador y el nombre que aparece en su tarjeta de crédito. 3. Para terminar la compra, debe hacer clic en el botón 'terminar'. 4. Los datos se envían a través de una conexión segura a una Terminal virtual. 5. La aplicación para manejo de transacciones con tarjetas de crédito se encarga de autorizar el pago. 6. Si el pago es rechazado se regresa a la pantalla de confirmación de pedido y se vuelve a mostrar el formulario de pago notificándole el error y sugiriéndole que elija otra tarjeta de crédito. 7. Se envía un correo electrónico al cliente notificándole que su pago fue aceptado y que su pedido está siendo surtido. 9. Se envía un correo al departamento de ventas para asignar un vendedor al nuevo pedido.
Excepciones:	El pago es rechazado, se da la opción de ingresar otro número de tarjeta. Sólo hay tres intentos.
Observaciones:	

ID del Caso de Uso:	10		
Nombre del Caso de Uso:	Iniciar Sesión		
Creado Por:	Svicente	Fecha de Creación:	07/11/2004

Actores:	Cliente Visitante, Cliente Comprador
Descripción:	El cliente proporciona su nombre de usuario y su contraseña si son válidos se inicia una sesión, si no se proporciona una alternativa dependiendo de la razón por la cual no fue autorizado.
Precondiciones:	El cliente debe haberse registrado con anterioridad para contar con un nombre de usuario y una contraseña.
Poscondiciones:	Cliente con una sesión iniciada en el sitio.
Flujo Normal:	<ol style="list-style-type: none"> 1. Se despliega un formulario con un cuadro de texto para el nombre de usuario, otro para la contraseña y un botón para enviar la información. 2. El cliente llena el formulario con los datos que eligió al momento de registrarse. 3. El sistema busca el nombre de usuario en la base de datos. 4. Una vez encontrado codifica la contraseña proporcionada y la compara con la almacenada en la base de datos. 5. Una vez comprobada la existencia de ambos datos, se crea una sesión para este cliente. 6. Su nombre de usuario y su id quedarán registrados en la sesión.
Flujos Alternos:	<ol style="list-style-type: none"> 4. Si el nombre de usuario no es encontrado en la base de datos, se muestra el formulario de inicio de sesión nuevamente y se despliega un mensaje de error sugiriéndole al cliente que teclee nuevamente sus datos. 5. Si el nombre de usuario existe en la base de datos pero la contraseña proporcionada no coincide, entonces se muestra el formulario de inicio de sesión y se despliega un mensaje de error sugiriéndole al cliente que teclee nuevamente sus datos. 6. Se muestra la opción de registrarse, pues probablemente sea un cliente visitante. 7. Se muestra una liga para la herramienta de recuperación de contraseñas.
Excepciones:	Si la sesión no se puede crear por cuestiones técnicas, tener una segunda opción par manejo de sesiones.
Observaciones:	

ID del Caso de Uso:	11		
Nombre del Caso de Uso:	Buscar un pedido en el sitio de e-commerce		
Creado Por:	Svicente	Fecha de Creación:	10/11/2004

Actores:	Cliente Comprador.
Descripción:	El Cliente puede buscar pedidos por fecha, por estado y por importe.
Precondiciones:	<ul style="list-style-type: none"> ▪ El Cliente Comprador debe haber iniciado una sesión en el sitio de e-commerce. ▪ Debe haber pedidos en la base de datos.
Poscondiciones:	Listado de pedidos.
Flujo Normal:	<ol style="list-style-type: none"> 1. Para los Clientes se muestra un formulario de búsqueda con los siguientes campos: <ol style="list-style-type: none"> a. Lista de estados disponibles (por definir). b. Fecha inicio (Lista día, mes, año). c. Fecha fin (Lista día, mes, año). d. Lista precio inicial. e. Lista precio final. f. Botón "Buscar". 2. Cuando el cliente da clic al botón "Buscar" se envían los parámetros y se hace una búsqueda combinada y sólo los resultados que cumplan con TODAS las condiciones y que pertenezcan a ese cliente. 3. Una vez realizada la consulta a la base de datos, se muestran los resultados en un listado con la fecha, estado e importe del pedido. Se muestra un botón "Ver detalles" al lado de cada renglón. También un botón "Imprimir".
Flujos Alternos:	<ol style="list-style-type: none"> 3. Si la consulta no regresó ningún resultado se muestra un mensaje de notificación. 4. Se muestra el formulario de búsqueda nuevamente. 5. Se listan los últimos 5 pedidos de ese cliente ordenados por fecha.
Excepciones:	Si el pedido no se encuentra en la base de datos, regresar al listado de todos los pedidos de ese cliente ordenados por fecha.
Observaciones:	

ID del Caso de Uso:	12		
Nombre del Caso de Uso:	Buscar un pedido en el sitio de Administración		
Creado Por:	Svicente	Fecha de Creación:	10/11/2004

Actores:	Vendedor, Almacenista.
Descripción:	El Actor puede buscar pedidos por fecha, por estado, por importe.
Precondiciones:	<ul style="list-style-type: none"> ▪ El vendedor y administrador deben haber iniciado una sesión en el sitio de administración. ▪ Debe haber pedidos en la base de datos.
Poscondiciones:	Listado de pedidos.
Flujo Normal:	<ol style="list-style-type: none"> 1. Se muestra un formulario de búsqueda con los siguientes campos: <ol style="list-style-type: none"> a. Lista de estados disponibles (por definir). b. Fecha inicio (Lista día, mes, año) y Fecha final (Lista día, mes, año). c. Botón "Buscar". 2. Cuando el actor da clic al botón "Buscar" se envían los parámetros y se hace una búsqueda combinada. 3. Sólo los resultados que cumplan con TODAS las condiciones se muestran en un listado con: la fecha, estado e importe del pedido, un botón para "Ver detalles y un botón para "Imprimir" el pedido.
Flujos Alternos:	<ol style="list-style-type: none"> 3. Si la consulta no regresó ningún resultado se muestra un mensaje de notificación. 4. Se muestra el formulario de búsqueda nuevamente.
Excepciones:	Si el pedido no se encuentra en la base de datos, regresar al listado.
Observaciones:	

ID del Caso de Uso:	13		
Nombre del Caso de Uso:	Buscar pedidos de un cliente en el sitio de Administración		
Creado Por:	Svicente	Fecha de Creación:	10/11/2004

Actores:	Vendedor, Almacenista.
Descripción:	El Actor puede buscar pedidos por Cliente.
Precondiciones:	<ul style="list-style-type: none"> ▪ El vendedor y administrador deben haber iniciado una sesión en el sitio de administración. ▪ Debe haber pedidos en la base de datos. ▪ El actor debe entrar al listado de clientes activos.
Poscondiciones:	Listado de pedidos de un cliente específico.
Flujo Normal:	<ol style="list-style-type: none"> 1. En el listado de clientes activos, el actor selecciona la opción "Ver pedidos para este Cliente". 2. Cuando el actor da clic al botón "Buscar" se envían el identificador de ese cliente y se realiza una consulta a la base de datos, buscando todos los pedidos de ese cliente. 3. Una vez realizada la consulta a la base de datos, se muestran los resultados en un listado con la fecha, estado e importe del pedido. Se muestra un botón "Ver detalles" al lado de cada renglón. También un botón "Imprimir".
Flujos Alternos:	<ol style="list-style-type: none"> 3. Si la consulta no regresó ningún resultado se muestra un mensaje de notificación. 4. Se regresa al listado de clientes.
Excepciones:	Si el cliente no tiene pedidos se regresa al listado de clientes y se envía un mensaje de notificación.
Observaciones:	

ID del Caso de Uso:	14		
Nombre del Caso de Uso:	Consultar detalles de un pedido		
Creado Por:	Svicente	Fecha de Creación:	10/11/2004

Actores:	Cliente Comprador, Vendedor, Almacenista
Descripción:	El actor selecciona de un listado de pedidos el que desea consultar. Después se muestra una ficha con los detalles de este pedido. Debe haber pedidos en la base de datos.
Precondiciones:	El Cliente Comprador debe haber iniciado una sesión en el sitio de e-commerce. El vendedor y administrador deben haber iniciado una sesión en el sitio de administración.
Poscondiciones:	Pedido consultado.
Flujo Normal:	<ol style="list-style-type: none"> 1. Se muestra un listado de pedidos con el número de pedido, la fecha, estado y el importe. Se muestra además un botón para "ver detalles". El listado es resultado de una búsqueda o el listado por defecto. 2. El actor hace clic en el botón "ver detalles" del pedido que desea consultar. 3. Se despliega una ficha con la fecha, estado, totales, datos personales del cliente, dirección de envío y los detalles del pedido (cantidad, producto, precio unitario e importe). 4. Se debe proporcionar alguna opción para cancelar dicho pedido. 5. En un panel independiente, pero en la misma página se muestra la información del envío del pedido.
Flujos Alternos:	
Excepciones:	Si el pedido no se encuentra en la base de datos, regresar al listado.
Observaciones:	

ID del Caso de Uso:	15		
Nombre del Caso de Uso:	Rastrea el envío de un pedido en el sitio de e-commerce		
Creado Por:	Svicente	Fecha de Creación:	10/11/2004

Actores:	Cliente Comprador
Descripción:	El actor selecciona consulta los detalles del pedido y ahí se muestra la información del envío de ese pedido.
Precondiciones:	<ul style="list-style-type: none"> ▪ El Cliente Comprador debe haber iniciado una sesión en el sitio de e-commerce. ▪ El Cliente Comprador debe tener pedidos. ▪ El Pedido debe haber sido enviado.
Poscondiciones:	Pedido consultado.
Flujo Normal:	<ol style="list-style-type: none"> 1. Se muestra un listado de pedidos con el número de pedido, la fecha, estado y el importe. Se muestra además un botón para "ver detalles". El listado es resultado de una búsqueda o el listado por defecto. 2. El actor hace clic en el botón "ver detalles" del pedido que desea consultar. 3. Se despliega una ficha con la fecha, estado, totales, datos personales del cliente, dirección de envío y los detalles del pedido (cantidad, producto, precio unitario e importe). 4. Mediante el identificador del pedido, se obtiene el número de guía del paquete enviado. 5. El sistema se conecta a una aplicación que proporciona el proveedor del servicio de envío y obtiene los datos de envío. 6. Esta información se presenta en un panel independiente en la misma página.
Flujos Alternos:	<ol style="list-style-type: none"> 3. Si no hay número de guía asignado quiere decir que el pedido no se ha enviado. 4. En el lugar donde debe aparecer el panel de envío se muestra un mensaje notificando que el pedido no ha sido enviado aún.
Excepciones:	Si el sistema del proveedor no está en línea, se muestra un mensaje de no disponibilidad.
Observaciones:	

ID del Caso de Uso:	16		
Nombre del Caso de Uso:	Rastrear el envío de un pedido en el sitio de administración		
Creado Por:	Svicente	Fecha de Creación:	10/11/2004

Actores:	Vendedor, Almacenista
Descripción:	El actor selecciona consulta los detalles del pedido y ahí se muestra la información del envío de ese pedido.
Precondiciones:	<ul style="list-style-type: none"> ▪ El Actor debe haber iniciado una sesión en el sitio de administración. ▪ El Pedido seleccionado debe existir. ▪ El Pedido debe haber sido enviado.
Poscondiciones:	Pedido consultado.
Flujo Normal:	<ol style="list-style-type: none"> 1. Se muestra un listado de pedidos con el número de pedido, la fecha, estado y el importe. Se muestra además un botón para "ver detalles". El listado es resultado de una búsqueda o el listado por defecto. 2. El actor hace clic en el botón "ver detalles" del pedido que desea consultar. 3. Se despliega una ficha con la fecha, estado, totales, datos personales del cliente, dirección de envío y los detalles del pedido (cantidad, producto, precio unitario e importe). 4. Mediante el identificador del pedido, se obtiene el número de guía del paquete enviado. 5. El sistema se conecta a una aplicación que proporciona el proveedor del servicio de envío y obtiene los datos de envío. 6. Esta información se presenta en un panel independiente en la misma página.
Flujos Alternos:	<ol style="list-style-type: none"> 5. Si no hay número de guía asignado quiere decir que el pedido no se ha enviado. 6. En el lugar donde debe aparecer el panel de envío se muestra un mensaje notificando que el pedido no ha sido enviado aún.
	<ol style="list-style-type: none"> 1. En la sección rastrear pedido, mostrar un formulario con un cuadro de texto para teclear la guía de embarque. 2. El sistema se conecta a una aplicación que proporciona el proveedor del servicio de envío y obtiene los datos de envío. 3. Esta información se presenta en una página de resultados.
Excepciones:	Si el sistema del proveedor no esta en línea, se muestra un mensaje de no disponibilidad.
Observaciones:	

ID del Caso de Uso:	17		
Nombre del Caso de Uso:	Devolver un pedido		
Creado Por:	Svicente	Fecha de Creación:	11/11/2004

Actores:	Cliente Comprador
Descripción:	El Cliente Comprador devuelve un pedido a la tienda.
Precondiciones:	<ul style="list-style-type: none"> ▪ El Cliente Comprador debe haber hecho una compra. ▪ El Pedido debe haber sido entregado al Cliente Comprador.
Poscondiciones:	<ul style="list-style-type: none"> ▪ Pedido cancelado. ▪ Orden de devolución.
Flujo Normal:	<ol style="list-style-type: none"> 1. El Cliente notifica al Vendedor las causas de la devolución. 2. El Vendedor llena un formulario de devolución. 3. A este formulario se anexa el pedido correspondiente. 4. Si los productos son devueltos por alguna falla se hace el diagnostico correspondiente, en base a éste se decide la acción a tomar (reparación, garantía proveedor). 5. El estado del Pedido cambia a Devuelto.
Flujos Alternos:	
Excepciones:	
Observaciones:	

ID del Caso de Uso:	18		
Nombre del Caso de Uso:	Actualizar datos del Cliente		
Creado Por:	Svicente	Fecha de Creación:	11/11/2004

Actores:	Cliente Comprador, Vendedor
Descripción:	El Cliente Comprador o el Vendedor actualizan los datos de un cliente.
Precondiciones:	<ul style="list-style-type: none"> ▪ El Cliente Comprador debe estar registrado. ▪ El Cliente Comprador o el Vendedor deben haber iniciado una sesión, ya sea en el sitio de e-commerce o en el administrador.
Poscondiciones:	<ul style="list-style-type: none"> ▪ Datos del Cliente actualizados.
Flujo Normal:	<ol style="list-style-type: none"> 1. El Cliente comprador entra a la sección "Mi cuenta" 2. Elige la opción "Actualizar datos". 3. Se despliega un formulario con los datos del cliente, domicilio particular y de envío. 4. El Cliente modifica sus datos. 5. Da clic en botón "Actualizar". 6. Los datos son enviados. 7. El sistema verifica que los datos sean válidos. 8. Los datos en la base de datos son actualizados.
Flujos Alternos:	<ol style="list-style-type: none"> 1. El Vendedor elige el Cliente a modificar. 2. Elige la opción "Actualizar datos". 3. Se despliega un formulario con los datos del cliente, domicilio particular y de envío. 4. El Vendedor modifica los datos del Cliente seleccionado. 5. Da clic en botón "Actualizar". 6. Los datos son enviados. 7. El sistema verifica que los datos sean válidos. 8. Los datos en la base de datos son actualizados.
	<ol style="list-style-type: none"> 8. Si alguno de los datos no es válido, se envía un mensaje indicando el campo en el que ocurrió el error.
Excepciones:	
Observaciones:	Para la validación se usan las mismas reglas que para el registro de clientes.

ID del Caso de Uso:	19		
Nombre del Caso de Uso:	Verificar pedidos pendientes		
Creado Por:	Svicente	Fecha de Creación:	11/11/2004

Actores:	Cliente Comprador, Vendedor, Almacenista
Descripción:	El sistema muestra todos los pedidos pendientes, ya sea solo para el cliente o todos en caso del Vendedor y Almacenista.
Precondiciones:	<ul style="list-style-type: none"> ▪ El Cliente Comprador debe estar registrado. ▪ El Cliente Comprador o el Vendedor deben haber iniciado una sesión, ya sea en el sitio de e-commerce o en el administrador. ▪ Debe haber Pedidos con estado Pendiente.
Poscondiciones:	Listado de Pedidos Pendientes.
Flujo Normal:	<ol style="list-style-type: none"> 1. Se realiza una búsqueda de todos los Pedidos con estado pendiente. Para el Cliente Comprador sólo se muestran aquellos suyos. 2. Para el Vendedor se muestran todos los Pedidos pendientes ordenados por fecha.
Flujos Alternos:	
Excepciones:	
Observaciones:	

ID del Caso de Uso:	20		
Nombre del Caso de Uso:	Solicitar una salida al almacén		
Creado Por:	Svicente	Fecha de Creación:	11/11/2004

Actores:	Vendedor
Descripción:	El Vendedor genera una solicitud de salida del almacén requiriendo los productos de un pedido nuevo.
Precondiciones:	<ul style="list-style-type: none"> ▪ El Vendedor debe haber iniciado una sesión en el sitio de administración. ▪ Debe haber seleccionado un pedido nuevo.
Poscondiciones:	Una nueva Solicitud de salida del almacén.
Flujo Normal:	<ol style="list-style-type: none"> 1. El Vendedor consulta los pedidos del día o los creados recientemente. 2. Selecciona el que el crea conveniente. 3. Hace clic en el botón "Generar Solicitud de Salida del Almacén". 4. El sistema genera una nueva Solicitud de Salida en la que se registra el número de pedido, los productos que componen el pedido con su cantidad y precio de venta, fecha y nombre de vendedor responsable. 5. Se envía una notificación al encargado del almacén para que surta la solicitud lo antes posible.
Flujos Alternos:	
Excepciones:	
Observaciones:	

ID del Caso de Uso:	21		
Nombre del Caso de Uso:	Registrar recepción del pedido en sitio de administración.		
Creado Por:	Svicente	Fecha de Creación:	11/11/2004

Actores:	Almacenista
Descripción:	El Almacenista cambia el estado del pedido a Entregado.
Precondiciones:	El Almacenista debe haber iniciado una sesión en el sitio de administración. Debe haber seleccionado un pedido.
Poscondiciones:	Pedido con estado de Entregado.
Flujo Normal:	<ol style="list-style-type: none"> 1. El Almacenista realiza una búsqueda para encontrar el pedido que al que quiere cambiar el estado. 2. Hace clic en el botón "Registrar recepción". 3. Se despliega una ficha con los datos principales del pedido (Número, Cliente, Importe, Identificador del Pedido). Y un cuadro de opción para marcar el pedido como Entregado. 4. Se envían los datos. 5. El estado se actualiza en la base de datos.
Flujos Alternos:	
Excepciones:	
Observaciones:	

ID del Caso de Uso:	22		
Nombre del Caso de Uso:	Enviar solicitud de Cancelación un pedido.		
Creado Por:	Svicente	Fecha de Creación:	11/11/2004

Actores:	Cliente Comprador
Descripción:	El Cliente Comprador selecciona un pedido y envía una solicitud para cancelarlo.
Precondiciones:	<ul style="list-style-type: none"> ▪ El Cliente Comprador debe haber iniciado una sesión en el sitio de e-commerce. ▪ El Cliente Comprador debe seleccionar el Pedido a Cancelar.
Poscondiciones:	Petición de cancelación enviada.
Flujo Normal:	<ol style="list-style-type: none"> 1. El cliente consulta sus pedidos. 2. Selecciona la opción "Ver detalles". 3. Hace clic en el botón "Cancelar este pedido". 4. Se muestra un formulario con un campo para anotar los motivos de la cancelación. 5. Hace clic en el botón "Enviar". 6. Una vez recibida por el sistema, el Pedido en cuestión, cambia de estado a "Cancelado por el Cliente." 7. Se envía un correo electrónico al vendedor responsable de despachar ese pedido.
Flujos Alternos:	
Excepciones:	
Observaciones:	

ID del Caso de Uso:	23		
Nombre del Caso de Uso:	Actualizar el estado de un pedido.		
Creado Por:	Svicente	Fecha de Creación:	11/11/2004

Actores:	Vendedor
Descripción:	El Vendedor cambia el estado de un Pedido
Precondiciones:	<ul style="list-style-type: none"> ▪ El Vendedor debe haber iniciado una sesión en el sitio de Administración. ▪ El Cliente Comprador debe seleccionar el Pedido para cambiar estado.
Poscondiciones:	Pedido con estado cambiado.
Flujo Normal:	<ol style="list-style-type: none"> 1. El Vendedor elige un pedido. 2. Selecciona la opción "Cambiar Estado". 3. Se muestra una ventana con un Listado de los Estatus disponibles. 4. Se selecciona el estado. 5. Hace clic en el botón "Enviar". 6. Una vez recibida por el sistema, el Pedido en cuestión, cambia de estado al estado seleccionado.
Flujos Alternos:	
Excepciones:	
Observaciones:	

ID del Caso de Uso:	24		
Nombre del Caso de Uso:	Mostrar promociones.		
Creado Por:	Svicente	Fecha de Creación:	11/11/2004

Actores:	Sistema
Descripción:	El Sistema muestra un listado de opciones.
Precondiciones:	<ul style="list-style-type: none"> ▪ Debe haber productos capturados. ▪ Debe haber promociones capturadas.
Poscondiciones:	Listado de Promociones.
Flujo Normal:	<ol style="list-style-type: none"> 1. El sistema busca los 10 registros más recientes en la tabla de promociones. 2. Se complementa el listado con los datos de cada producto asociado a una promoción.
Flujos Alternos:	
Excepciones:	
Observaciones:	

ID del Caso de Uso:	25		
Nombre del Caso de Uso:	Recuperar contraseña.		
Creado Por:	Svicente	Fecha de Creación:	11/11/2004

Actores:	Cliente Comprador
Descripción:	Si el Cliente Comprador ha extraviado su contraseña, puede recuperarla proporcionando su nombre de usuario, su correo electrónico y la respuesta a la pregunta secreta.
Precondiciones:	El Cliente Comprador debe estar registrado.
Poscondiciones:	Contraseña del Cliente enviada por correo.
Flujo Normal:	<ol style="list-style-type: none"> 1. El Cliente Comprador entra a la sección "Recuperar Contraseñas". 2. Se despliega un formulario con un cuadro de texto para el nombre de usuario, correo electrónico. Se muestra la pregunta secreta y un cuadro de texto para introducir la respuesta. Además se muestra un botón para enviar los datos. 3. Se busca en la base de datos los registros que correspondan a los datos: nombre de usuario, correo electrónico y respuesta. 4. Si se encontró un registro, se genera una contraseña aleatoria compuesta de letras y números únicamente, 5. Se codifica la contraseña y se actualiza en la base de datos. 6. Se envía un correo electrónico al cliente con su nombre de usuario y la nueva contraseña.
Flujos Alternos:	<ol style="list-style-type: none"> 4. Si no se encontró ningún registro, se muestra un mensaje indicándole que los datos no se encontraron en la base de datos. 5. Se muestra un vínculo para ir a la sección "Registro de Clientes". 6. Se muestra nuevamente el formulario.
Observaciones:	Bloquear el número de intentos y el número de veces que se puede recuperar la contraseña.

ID del Caso de Uso:	26		
Nombre del Caso de Uso:	Recibir solicitud de salida de almacén		
Creado Por:	gvelasco	Fecha de Creación:	27-05-2004

Actores:	Almacenista
Descripción:	Permite verificar solicitudes salida de almacén
Precondiciones:	<ul style="list-style-type: none"> ▪ El almacenista debe estar firmado en el sistema ▪ Debe existir un pedido por surtir. ▪ El pedido debe tener un folio asignado
Poscondiciones:	1. El pedido es verificado por el almacenista
Flujo Normal:	<ol style="list-style-type: none"> 1. El almacenista verifica pedido hecho por cliente comprador. 2. Surte pedido 3. Despacha pedido de orden y la asigna a porteador
Flujos Alternos:	El almacenista no cuenta con existencias en almacén.
Excepciones:	Existencias no han sido actualizadas en el sistema.

ID del Caso de Uso:	27		
Nombre del Caso de Uso:	Actualizar entradas/salidas almacén		
Creado Por:	gvelasco	Fecha de Creación:	27-05-2004

Actores:	Almacenista
Descripción:	Actualiza entradas/salidas almacén
Precondiciones:	<ol style="list-style-type: none"> 1. El almacenista debe estar firmado en el sistema 2. Deben existir artículos por ingresar o retirar del almacén. 3. Los artículos deben tener todas las características para ser ingresados o dados de baja del almacén.
Poscondiciones:	<ol style="list-style-type: none"> 4. Los artículos son verificados por el almacenista
Flujo Normal:	<ol style="list-style-type: none"> 4. El almacenista recibe artículos por ingresar o retirar del almacén. 5. Realiza cambios en el sistema 6. Hace ingresos o retiros de mercancía físicamente
Flujos Alternos:	El almacenista no cuenta con las categorías necesarias para dar de alta productos.
Excepciones:	Existencias no han sido actualizadas de manera correcta.

ID del Caso de Uso:	28		
Nombre del Caso de Uso:	Entrega Pedido a porteador		
Creado Por:	gvelasco	Fecha de Creación:	27-05-2004

Actores:	Almacenista
Descripción:	Entrega pedido a porteador
Precondiciones:	<ul style="list-style-type: none"> ▪ El almacenista debe estar firmado en el sistema ▪ El almacenista debe haber seleccionado empresa de mensajería ▪ El almacenista debe haber asignado al número de pedido el número de guía.
Poscondiciones:	<ul style="list-style-type: none"> ▪ Paquete es entregado a empresa de mensajería
Flujo Normal:	<ol style="list-style-type: none"> 1. El almacenista prepara pedido para ser entregado 2. Selecciona empresa de mensajería 3. Asigna pedido a número de guía 4. Entrega pedido a porteador 5. Da seguimiento 6. Da de baja total de almacén
Flujos Alternos:	Da de baja del almacén antes de ser notificado
Excepciones:	No hay empresa de mensajería para hacer la entrega.

ID del Caso de Uso:	29		
Nombre del Caso de Uso:	Actualiza entradas/salidas almacén		
Creado Por:	gvelasco	Fecha de Creación:	27-05-2004

Actores:	Responsable del catálogo de productos
Descripción:	Actualiza entradas/salidas de productos
Precondiciones:	<ul style="list-style-type: none"> ▪ El responsable debe estar firmado en el sistema ▪ Deben existir artículos por ingresar o retirar del almacén. ▪ Los artículos deben tener todas las características para ser ingresados o dados de baja en el sistema.
Poscondiciones:	<ul style="list-style-type: none"> ▪ Los artículos son verificados por el responsable
Flujo Normal:	<ol style="list-style-type: none"> 1. El responsable recibe artículos por ingresar o retirar del catálogo. 2. Realiza cambios en el catálogo 3. Hace altas o bajas de artículos
Flujos Alternos:	El responsable puede no recibir los datos para ingresarlos.
Excepciones:	El responsable no cuenta con los datos completos para dar de alta los productos en el catálogo.

ID del Caso de Uso:	30		
Nombre del Caso de Uso:	Dar de alta usuarios del sistema		
Creado Por:	Gvelasco	Fecha de Creación:	27-05-2004

Actores:	Administrador de Sistema
Descripción:	Dar de alta usuarios
Precondiciones:	<ul style="list-style-type: none"> ▪ El administrador debe estar firmado en el sistema ▪ Debe existir al menos una alta por realizar. ▪ El administrador debe tener los datos completos de la persona a registrar como alta. ▪ El usuario a registrar debe contar con la autorización de jefe inmediato para tener acceso a la firma. ▪ El administrador debe contar con el nivel de acceso a otorgar al usuario.
Poscondiciones:	El administrador debe verificar el acceso otorgado al usuario
Flujo Normal:	<ol style="list-style-type: none"> 1. El administrador recibe los datos del usuario a dar de alta. 2. Realiza cambios en el sistema 3. Hace ingresos de información y registra altas de usuario
Flujos Alternos:	El administrador no cuenta con los datos necesarios para dar de alta al usuario en el sistema.
Excepciones:	El usuario no ha sido dado de alta con los permisos correspondientes.

ID del Caso de Uso:	31		
Nombre del Caso de Uso:	Dar de baja usuarios		
Creado Por:	gvelasco	Fecha de Creación:	27-05-2004

Actores:	Administrador de sistema
Descripción:	Da de baja cuentas activas en el sistema
Precondiciones:	<ul style="list-style-type: none"> ▪ El administrador del sistema debe tener un requerimiento de baja de cuenta ▪ La cuenta debe existir ▪ La cuenta debe estar activa
Poscondiciones:	Se verifica que la baja haya sido realizada de manera correcta
Flujo Normal:	<ol style="list-style-type: none"> 1. El administrador recibe requerimiento de baja de cuenta. 2. El administrador verifica que exista la cuenta en el sistema y que esté activa. 3. El administrador realiza la baja lógica de la cuenta del sistema.
Flujos Alternos:	El administrador encuentra que la cuenta no está activa o que no existe.
Excepciones:	La cuenta no se dio de baja de manera correcta.

ID del Caso de Uso:	32		
Nombre del Caso de Uso:	Cambios a las cuentas de usuarios		
Creado Por:	gvelasco	Fecha de Creación:	27-05-2004

Actores:	Administrador de sistema
Descripción:	Modifica las propiedades de las cuentas activas de usuarios
Precondiciones:	<ul style="list-style-type: none"> ▪ El administrador del sistema debe tener un requerimiento de modificación de cuenta ▪ La cuenta debe existir ▪ La cuenta debe estar activa
Poscondiciones:	Se verifica que el cambio de propiedades haya sido realizado de manera correcta
Flujo Normal:	<ol style="list-style-type: none"> 1. El administrador recibe requerimiento de cambio de cuenta. 2. El administrador verifica que exista la cuenta en el sistema y que esté activa. 3. El administrador realiza las modificaciones solicitadas a la cuenta del sistema.
Flujos Alternos:	El administrador encuentra que la cuenta no está activa o que no existe.
Excepciones:	La cuenta no pertenece al perfil correspondiente de los cambios solicitados.

ID del Caso de Uso:	33		
Nombre del Caso de Uso:	Crear categorías de productos		
Creado Por:	gvelasco	Fecha de Creación:	27-05-2004

Actores:	Administrador de sistema
Descripción:	Da de alta las nuevas categorías de producto
Precondiciones:	<ol style="list-style-type: none"> 1. El administrador del sistema debe tener un requerimiento de alta de categoría de producto 2. Debe tener todos los datos requeridos para dar de alta una categoría de producto.
Poscondiciones:	Se verifica que se haya dado de alta la categoría de producto de manera correcta
Flujo Normal:	<ol style="list-style-type: none"> 1. El administrador recibe requerimiento de alta de categoría de producto. 2. El administrador verifica que exista la información correspondiente a la nueva categoría de sistema 3. El administrador realiza los movimientos necesarios para crear el alta de categoría.
Flujos Alternos:	El administrador encuentra que le hacen falta datos para poder dar de alta la categoría.
Excepciones:	La categoría no cuenta con las características para ser considerada como tal.

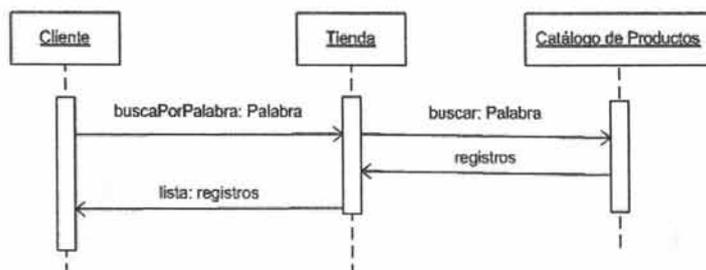
ID del Caso de Uso:	34		
Nombre del Caso de Uso:	Asignar permisos a las cuentas de usuario		
Creado Por:	gvelasco	Fecha de Creación:	27-05-2004

Actores:	Administrador de sistema
Descripción:	Modifica las propiedades de las cuentas activas de usuarios
Precondiciones:	<ul style="list-style-type: none"> ▪ El administrador del sistema debe tener un requerimiento de modificación de cuenta ▪ La cuenta debe existir ▪ La cuenta debe estar activa
Poscondiciones:	Se verifica que el cambio de propiedades haya sido realizado de manera correcta
Flujo Normal:	<ol style="list-style-type: none"> 1. El administrador recibe requerimiento de cambio de cuenta. 2. El administrador verifica que exista la cuenta en el sistema y que esté activa. 3. El administrador realiza las modificaciones solicitadas a la cuenta del sistema.
Flujos Alternos:	El administrador encuentra que la cuenta no está activa o que no existe.
Excepciones:	La cuenta no pertenece al perfil correspondiente de los cambios solicitados.

2.5.4 Diagramas de secuencia

A continuación se muestran algunos de los diagramas de secuencia para el caso práctico de este trabajo.

Búsqueda de productos por palabra



Agregar productos al carrito

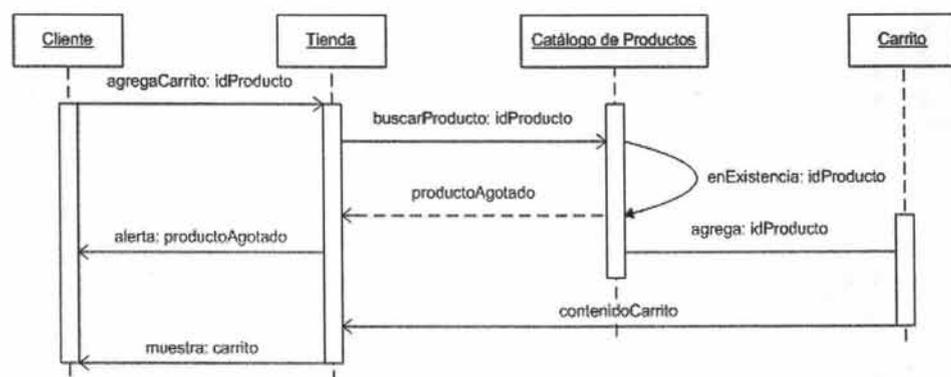


Figura 2.5.4 Diagramas de caso de uso

2.6 Mejores prácticas aplicadas al diseño de sistemas

En este apartado del diplomado, los instructores, en base a su experiencia profesional, nos proporcionaron algunas metodologías para aplicar en el desarrollo de nuestro sistema web.

Desarrollo iterativo de software

Sugiere que se deben generar versiones mejoradas del sistema, es decir, se va generando un sistema parcialmente funcional que se va refinando con la evaluación del usuario y con base en requerimientos.

Arquitectura basada en componentes

Recomienda la reutilización de componentes.

Modelado visual de software

Permite la relación consistente entre diseño e implementación.

Verificación de la calidad del software

El sistema debe someterse a pruebas, pero no por parte del equipo de desarrolladores. Idealmente debe existir un equipo para pruebas, conformado por usuarios comunes del sistema quienes son los que finalmente determinan si el sistema cumple o no con su propósito.

Documentación de cambios

Todo cambio debe ser registrado y evaluado antes de llevarlo a cabo.

Control de Versiones

Por cada versión liberada debe existir la documentación que especifique los cambios significativos y mejoras de la versión.

Flujo de trabajo

Secuencia de actividades a seguir para obtener una respuesta a una solicitud del sistema.

Capítulo 3

Java como herramienta de desarrollo, un enfoque orientado a objetos.

3.1 Características generales de Java

Java¹ fue ideado en 1990 como un lenguaje para controlar dispositivos electrónicos de consumo, era conocido como Oak, roble en inglés, para denotar que se trataba de un lenguaje robusto. Sin embargo no tuvo el éxito esperado y se abandono el proyecto.

Para 1995 fue retomado, ahora con el nombre de Java y se le dio un enfoque diferente para poder ser competitivo.

Este lenguaje nos ofrece las siguientes bondades:

- Simple
 - o No maneja punteros
 - o No es complejo
 - o No maneja directamente objetos en memoria
 - o No hay herencia múltiple
- Orientado a objetos
- Distribuido
- Interpretado
- Robusto
 - o Verificación de índices de arreglos.
 - o Verificación de tipo de datos durante la ejecución o interpretación.
 - o Garbage collector (Limpieza de la referencias a objetos en memoria)
- Seguro
 - o Detección de código ilegal
 - o Ejecución de aplicaciones en red.

¹ Ver <http://www.java.sun.com>

- No puede acceder a memoria.
- Autenticación de archivos enviados por la red.
- Restricción de programas de red, archivos y aplicaciones.
- Multiplataforma
 - Necesita una Máquina Virtual para interpretar el código, existe una máquina para cada plataforma.
 - Compatibilidad de librerías, fuentes y código.
- Portable
 - Mismos tipos de datos en todas las plataformas.
 - Sistema abstracto de ventanas.
- Multithread
 - Varios programas ejecutado al mismo tiempo.
- Dinámico
 - Los objetos son cargados según se van requiriendo.
- Es un lenguaje de programación y plataforma a la vez.
 - El compilador traduce el programa a bytecodes (*.class)
 - El intérprete analiza gramaticalmente y ejecuta cada instrucción línea por línea.
 - La única desventaja es que la ejecución es un poco más lenta.

3.2 La plataforma Java

3.2.1. Máquina virtual

La Máquina Virtual Java (JVM) es el entorno en el que se ejecutan los programas Java, su misión principal es la de garantizar la portabilidad de las aplicaciones Java.

Define esencialmente un motor abstracto y especifica las instrucciones (*bytecodes*) que este motor puede ejecutar. El intérprete Java específico ejecuta

las instrucciones que se guardan en los archivos cuya extensión es .class. Las tareas principales de la JVM son las siguientes:

- Reservar espacio en memoria para los objetos creados
- Liberar la memoria no usada (garbage collection).
- Asignar variables a registros y pilas
- Llamar al sistema huésped para ciertas funciones, como los accesos a los dispositivos
- Vigilar el cumplimiento de las normas de seguridad de las aplicaciones Java

La siguiente figura muestra de manera gráfica la lógica de la ejecución de un programa a través de la máquina virtual.

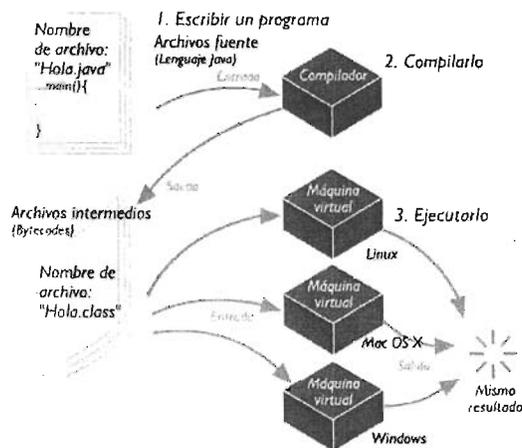


Figura 3.2.1 Ejecución de un programa a través de la máquina virtual

3.2.2. API

Es una colección de componentes y clases reutilizables agrupados en paquetes según su utilidad.

- Básicas
- Applets

- Red
- Internacionalización
- Seguridad
- Java beans
- Señalización
- JDBC

3.3 Tipos de aplicaciones

- Standalone
- Applets

Reside en un servidor con bytecodes, se accede a través de un navegador web. Se ejecutan localmente.

- Servlets

La aplicación se ejecuta en el servidor, el cliente solo recibe la salida en html.

- Jsp

Se accede a través del navegador web. Busca yn JSP, lo convierte en servlet y envía una respuesta en html.

Una de las desventajas de servlets y jsp es la funcionalidad que se pierde en la interfaz de usuario, ya que es html puro.

- Java beans

Son clases de java y permiten tener objetos reutilizables.

- o EJB

Clases de java. Componentes deplorables, necesitan de un contenedor.
Permite la división de trabajo a través de la especialización de equipos.

3.4 JDK (Java Developer Kit)

El Java Development Kit permite desarrollar una aplicación en Java. Está compuesto por:

- Compilador
- JVM
- API
- Utilerías

Se descarga del sitio web de Sun <http://java.sun.com>

Después de instalarlo se deben modificar las variables de ambiente Path y Classpath, agregando los directorios donde residen las clases.

3.5 Entorno de desarrollo Eclipse

Aplicación Open Source para programación en diversos lenguajes como Java.
Algunas características:

- JDT es la aplicación principal.
- Requiere de JRE y JVM.
- Plugins.
- Workspace. Mapeo de proyectos.
- Workbench. Interfaz de usuario (Vistas, editores, perspectivas).

3.6 *Estándar de codificación*

Clases: Clase ó MiClase

Interfaces: Interfaz ó MiInterfaz

Métodos: metodo() ó metodoLargo()

Métodos de acceso: getAtributo(), setAtributo()

Variables: altura o alturaMedia

Constantes: CONSTANTE ó CONSTANTE_LARGA

Paquetes: java.paquete.subpaquete

3.7 *Calidad y procesos de software*

Se debe tener en mente que la aplicación debe cumplir con los requerimientos del cliente. Para lograr esto, el problema debe haberse entendido en su totalidad.

Antes de codificar, se debe planear que es lo que se va a programar. Después diseñar la estructura del programa: ciclos, condiciones, saltos, etc. Todo esto en lenguaje común. Después se puede proceder a la codificación.

Entre cada etapa debe haber una revisión. El producto de la codificación deberá ser sometido a pruebas para la detección de errores y si el proceso se ha seguido, el tiempo de detección debe ser muy corto y el número de fallas mínimo.

Capítulo 4

Interfaz de usuario y sus aplicaciones.

4.1 La importancia de una buena interfaz

Una vez que el sistema Web ha sido analizado y modelado, se cuenta con los documentos necesarios para comenzar con el desarrollo de un prototipo.

Se trata de una vista a lo que será el sistema, pero aún sin funcionalidad alguna. El prototipo debe dar una idea clara de la manera en la que el usuario va a interactuar con el sistema a través de una interfaz gráfica.

Si durante la etapa de análisis y diseño se obtienen las bases de un sistema robusto, escalable y seguro, la interfaz de usuario debe ser lo suficientemente buena para que a los ojos del usuario, todo esto que está detrás, sea transparente y se logre cumplir con sus expectativas.

Una interfaz de usuario mal diseñada puede costar lo mismo o más que un análisis deficiente del sistema.

El primer documento fundamental que se obtiene es la definición de las características de los usuarios del sistema, después se desarrolla el mapa de navegación del sitio Web.

El siguiente paso es definir un documento con los requerimientos de la interfaz gráfica con base a las políticas de la empresa y su identidad gráfica.

Se prosigue a diseñar un bosquejo de la página de inicio y una página interior del sitio Web, en este caso mediante el lenguaje html y al uso de la herramienta de diseño seleccionada (Macromedia Dreamweaver).

Resumiendo, para poder pasar a la etapa de desarrollo de un prototipo se requiere tener definidos los siguientes elementos:

- Diagramas de casos de uso.
- Diagramas de secuencia.
- Identidad gráfica.
- Políticas.
- Estudio de mercado.

**ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA**

4.2 Niveles de usuario

En los diagramas de casos de uso, están definidos los actores y en qué etapa del sistema entran en acción, con base a esto se pueden comenzar a definir las interfaces para cada uno de ellos e identificar aquellas que serán comunes para todos. Aunque se trate de la misma acción, no todos los actores tienen los mismos privilegios. Algunas de las actividades que puede llevar a cabo un usuario son:

- Consulta básica o de área
- Consulta total
- Consulta y operaciones básica o de área
- Realización de operaciones totales
- Consulta, operaciones y generación de reportes de área
- Acceso total

Todas esas acciones se obtienen de los diagramas de secuencia, que también son útiles para determinar la secuencia de las pantallas para cada usuario.

Por regla general no debe haber usuarios con todos esos permisos, salvo el administrador del sistema.

Si por alguna razón se hace un mal diseño de los niveles de usuario se puede estar comprometiendo la seguridad del sistema.

Para reducir este riesgo, es preferible manejar un número grande de usuarios aunque implique el diseño de más pantallas.

4.3 Mapa de navegación

La finalidad de un mapa de navegación es no perder de vista la relación entre pantallas al momento de dar mantenimiento y administrar el sitio.

El mapa de navegación es una representación en forma de árbol de todas las pantallas que conforman el sitio Web. Los niveles indican que tan particular es la pantalla, es decir, a mayor nivel la funcionalidad es exclusiva para cierto tipo de usuario.

Al conformar el mapa se debe trabajar en un módulo a la vez, esto es, desarrollar rama por rama, nunca en paralelo. Si no se sabe el contenido de alguna de las pantallas, de cualquier manera debe ser tomada en cuenta en el mapa.

La estructura debe estar lo más equilibrada posible, esto es, agrupando procesos similares o que tengan el mismo número de pasos. Este equilibrio hace que el sitio sea más fácil de mantener.

4.4 Normatividad

Son lineamientos especificados por el cliente de manera escrita que cubren aspectos como identidad gráfica, logotipos, restricciones de color, políticas y todo aquello que el cliente considere importante de acuerdo a sus intereses.

Como desarrolladores se debe respetar, en la medida que sea posible, y en caso contrario justificar el porqué se van a ignorar algunas reglas. Siempre debe existir la posibilidad de negociar.

4.5 Bosquejo de diseño

El bosquejo de diseño es solamente una guía de lo que será la interfaz gráfica del sitio, basado en la normatividad y se plasma en papel solamente.

Es recomendable comenzar por el bosquejo de la página principal y de ahí hacer lo propio para dos niveles más.

Básicamente se establecen los siguientes puntos en este bosquejo:

- Elementos
- Orden
- Continuidad

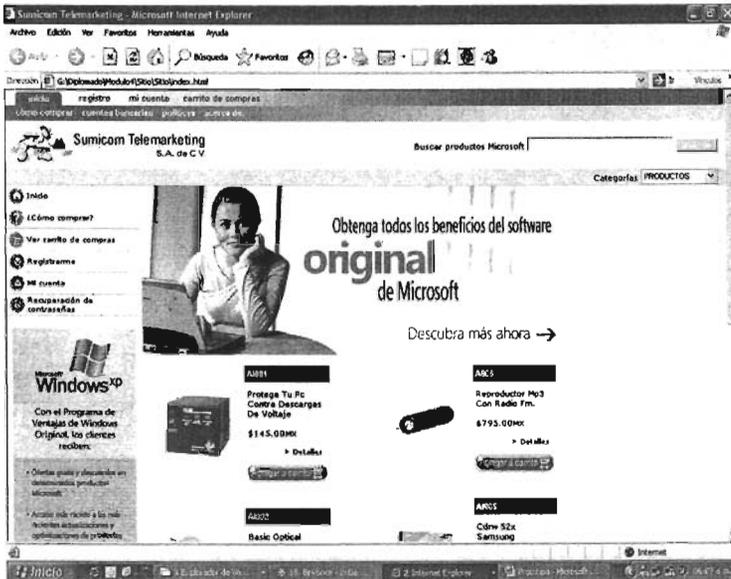
4.6 Prototipo

El prototipo es el siguiente paso a dar, después de que el usuario ha dado su aprobación al bosquejo. Es una visualización del sistema con cierto grado de interactividad simulada.

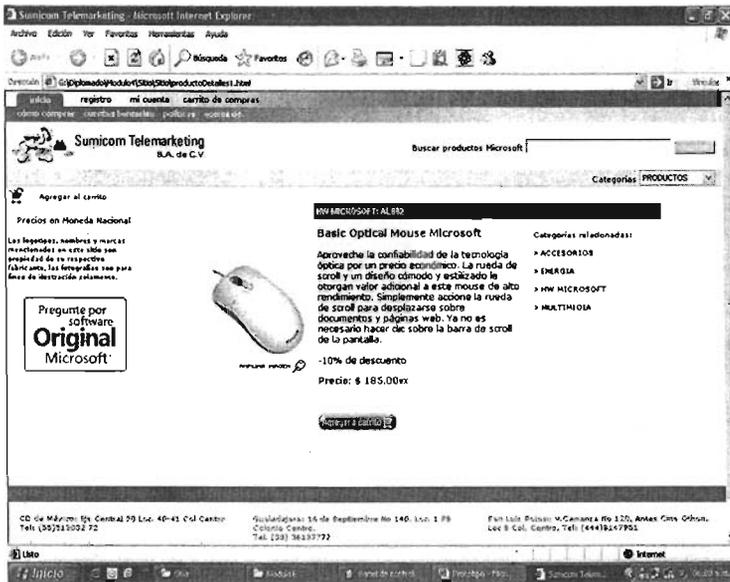
El prototipo deber ser evaluado por el usuario de viva voz para tomar nota de las observaciones y comprobar que efectivamente se cubrieron sus requerimientos.

Es en esta etapa en la que se detectan los errores en los perfiles de usuario.

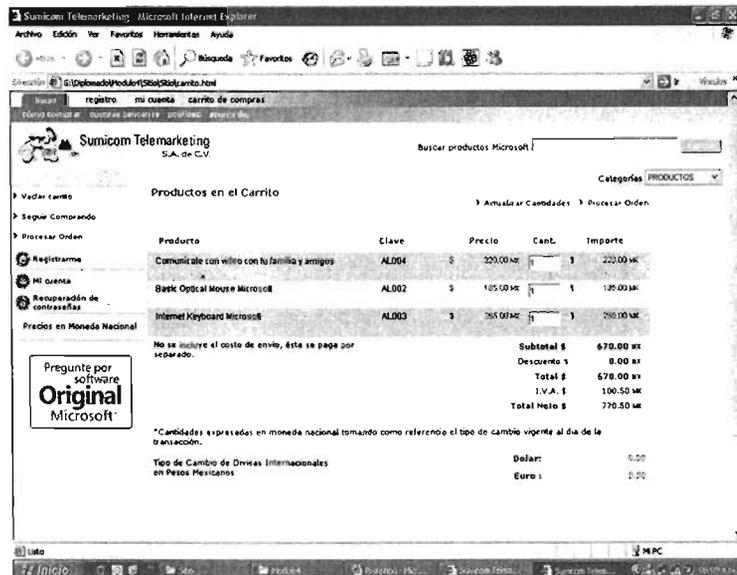
A continuación se muestran algunas de las pantallas del prototipo propuesto para el caso práctico de sumitel.



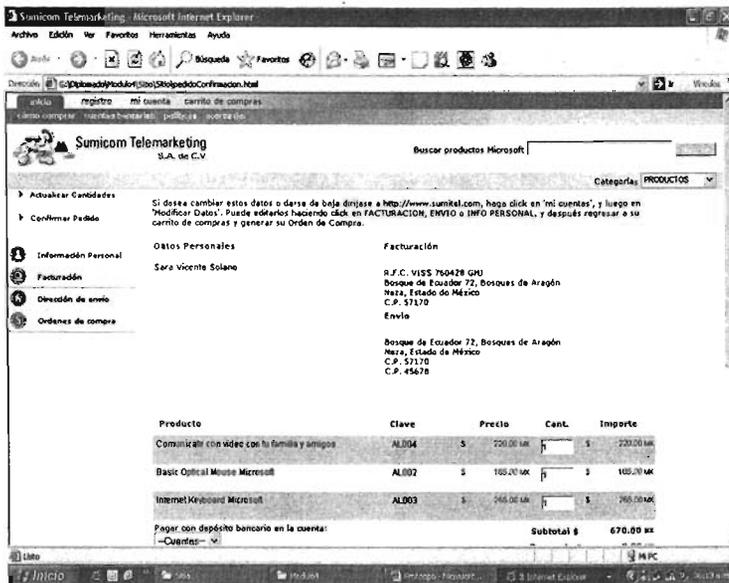
4.6.1. Página principal de la tienda virtual.



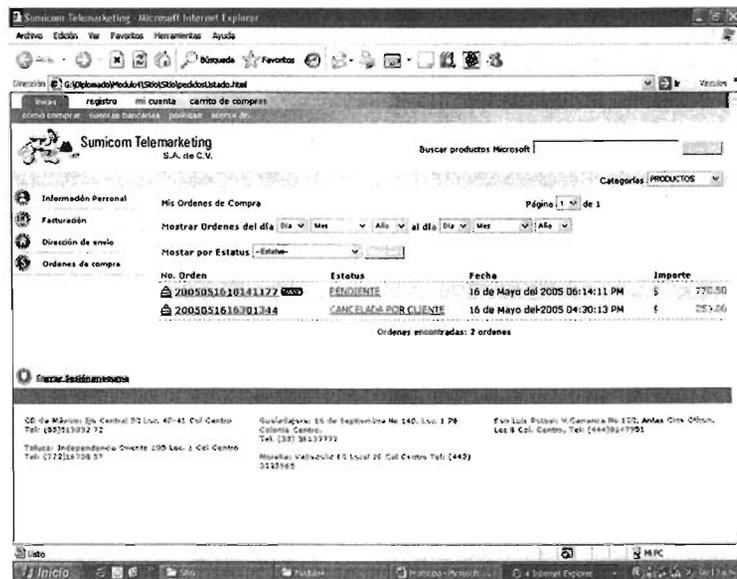
4.6.2. Detalles de un producto.



4.6.3. Vista del carrito de comprar con productos.



4.6.4. Vista de un pedido antes de confirmarse.



4.6.5. Listado de pedidos para un cliente.

Sumicom Telemarketing
S.A. de C.V.

registro mi cuenta centro de compras

Buscar productos Microsoft

Información personal Registro de Nuevos Clientes

Facturación Para ser cliente registrado solo necesitas llenar los siguientes tres formularios. Antes de enviar cualquier información, te sugerimos leer nuestras Políticas de venta y de Garantía.

Selección de envío

1: Datos Personales
Los ítems en negrita indican la información es requerida.

Login Contraseña

Confirme su contraseña Confirmar Contraseña

Truco Nombre

Apellido paterno Apellido materno

Correo Teléfono móvil

Fecha de nacimiento de de

Recomendaciones para la selección de tu nombre de usuario y contraseña.

No escribas tu contraseña, ni la digas cerca de su computadora.

El nombre de usuario y la contraseña debe contener de 4 a 20 caracteres que pueden ser solo letras y números.

No uses palabras comunes que puedan ser adivinadas con facilidad.

Camble tu contraseña cada 45 a 60 días.

Menos como la misma contraseña una vez de 90 días

4.6.6. Primera pantalla para registro de un cliente.

4.7 Evaluación

El usuario evalúa si sus requerimientos de han cubierto y que exista congruencia entre los elementos.

4.7.1. Usabilidad

En esta etapa comienza formalmente la construcción de la interfaz, tomando en cuenta ciertos aspectos:

Diseño amigable y atractivo

Asegurarse que los fondos de las páginas sean de un color neutro.

Perseguir el perfil de usuario.

Emplear imágenes en áreas estratégicas

No exceder los 800 x 600 pixeles para las pantallas.

Mantener siempre un orden y distribución en los elementos de las pantallas.

Congruencia de procesos

Los elementos de una pantalla, especialmente los formularios debe tener una distribución lógica, de acuerdo a lo definido en el diagrama de secuencia y respetando las políticas.

Navegación lógica

Este punto se refiere al orden y distribución en que son recorridos los elementos de una pantalla. Los elementos deben ser colocados de tal manera que al usuario le resulte fácil moverse a través de ellos. Por ejemplo, en un formulario con muchos campos, tal vez sea mejor subdividir la información en varias pantallas.

Retroalimentación

El usuario debe saber en que sección del sitio o paso de de un proceso se encuentra. Si las pantallas son muy similares entonces se recomienda:

- Indicar el paso o sección de la pantalla.
- Describir brevemente el proceso.
- Indicar condiciones o restricciones.
- Sugerencias.
- Eventos.
- Políticas y/o ayuda.

Implementación de la ayuda

La ayuda debe ir de acuerdo al nivel de usuario.

4.8 Herramientas de diseño

4.8.1. *Html*

Es el lenguaje más utilizado para el desarrollo de interfaces por su simplicidad, compatibilidad y rapidez.

Toda respuesta obtenida del sistema estará dada en html basándose en alguna de las alternativas de diseño:

- Tablas
- Marcos
- Css
- Plantillas
- Capas

Un ejercicio para determinar que alternativa tomar, fue realizar dos propuestas diferentes, comparar el resultado final y elegir una estrategia para todo el sitio.

Además se revisaron diversos ejemplos de formularios y las características esenciales de sus componentes.

4.8.2. *Macromedia™ Dreamweaver™*

La herramienta de trabajo seleccionada fue Macromedia™ Dreamweaver™ que integra un editor visual de html, editor de código y diversas funcionalidades para conexión con servidores remotos.

Cuenta con un administrador de sitios que, entre otras cosas, permite mantener el contenido lo más consistente posible ya que actualiza los vínculos entre los elementos del sitio (páginas, imágenes, scripts).

4.8.3. *Javascript*

Siguiendo con la interactividad con el usuario, el siguiente punto fue dar una introducción a javascript. Es un lenguaje que se ejecuta del lado del cliente y es una herramienta muy útil principalmente para validación de formularios.

4.9 Contenido dinámico

Se analizó la forma en que los jsp y java servlets interactúan con el usuario. En el módulo 1 ya se había llevado a cabo la publicación de una pequeña aplicación en java para ver este comportamiento.

Capítulo 5

Fundamentos y aplicaciones de bases de datos.

5.1 La capa de datos

Dentro de la arquitectura de 3 capas, la capa de datos está conformada por los datos que alimentarán al sistema web. Existen dos sistemas en esta capa: uno llamado gestor de bases de datos y otro que se encarga del acceso a ellos.

Dentro de una base de datos podemos diferenciar 3 conceptos:

- Los datos que físicamente están almacenado en un archivo en alguna parte del servidor web.
- El modelo de datos, es decir, el concepto de los datos, sus relaciones y restricciones.
- El esquema de la base de datos, que es la definición de los datos representados en tablas formadas por tuplas o registros que, a su vez, están formados por campos.
- El estado de una base de datos que indica como va cambiando a través del tiempo.

5.2 Sistema de gestión de bases de datos

Es un conjunto de aplicaciones que permiten definir, crear y mantener una base de datos. Una de esas aplicaciones se encarga de restringir el acceso a la base de datos.

Un gestor de bases de datos tiene tres niveles: el de vistas, el lógico y el físico.

El nivel de vistas proporciona diferentes maneras de mostrar los datos almacenados.

En el nivel lógico se encuentra el diseño conceptual de la base de datos definido por el lenguaje de definición de datos y el esquema.

El nivel físico los conforman los archivos que se almacenan en dispositivos de almacenamiento secundario.

Un buen sistema de gestión de bases datos debe asegurar la independencia de los datos, esto es, al modificar la estructura lógica no se debe afectar la parte física y viceversa.

5.3 Usuarios en el SGBD

Tenemos 4 niveles de usuario de una base de datos:

1. Usuarios comunes
2. Usuarios expertos
3. Programadores
4. Administradores (DBA)

El nivel de seguridad para los usuarios comunes es el más holgado, ya que se trata de usuario con muy pocos conocimientos en la administración de bases de datos.

En el caso de los usuarios expertos el nivel de seguridad es mayor ya que tienden a buscar información a través de consultas.

Para los programadores sólo se deben asignar privilegios para generar vistas solamente.

El DBA tiene el nivel de seguridad más alto ya que es quien mantiene la base de datos y conoce la estructura de la misma.

5.4 Funciones de un SGDB

- Almacenar los datos y su descripción.
- Asegurar la integridad de los datos.
- Manejo de concurrencia.
- Es función del SGBD verificar los procesos de todas las transacciones¹, así como garantizar la recuperación de los datos de la última versión óptima de la transacción en caso de algún desastre informático.

¹ Una transacción es un conjunto de operaciones que no se considera terminada hasta que la última instrucción se ha efectuado con éxito.

- Debe administrar de manera óptima la memoria y espacio de trabajo, además de entenderse directamente con el sistema de archivos.
- Establecer los niveles de seguridad.
- Establecer comunicación con sistemas externos.

5.5 Características deseables de una base de datos

- Versatilidad para presentar la información.
- Buen desempeño.
- Redundancia mínima.
- Simplicidad.
- Seguridad.
- Privacidad.
- Afinación.
- Integridad.

5.6 Modelo relacional de bases de datos

Establece ciertas reglas basadas en la teoría matemática de conjuntos. Un conjunto es una serie de elementos de una naturaleza similar.

Una tabla puede ser vista como un conjunto, por lo tanto, se pueden aplicar las mismas operaciones.

Las bases de datos en este modelo se representan con un diagrama de Entidad-Relación.

Durante el módulo 2 ya se había cubierto esta parte, lo que se hizo en este módulo fue retomarlo que ya se había hecho y complementarlo.

Por ejemplo, se tenían ya definidas las entidades, pero no las relaciones y durante el proceso de identificación de estas surgieron algunos ajustes al diagrama.

Con base a estos ajustes la estructura de la base de datos queda como se muestra en la figura 5.6.1.

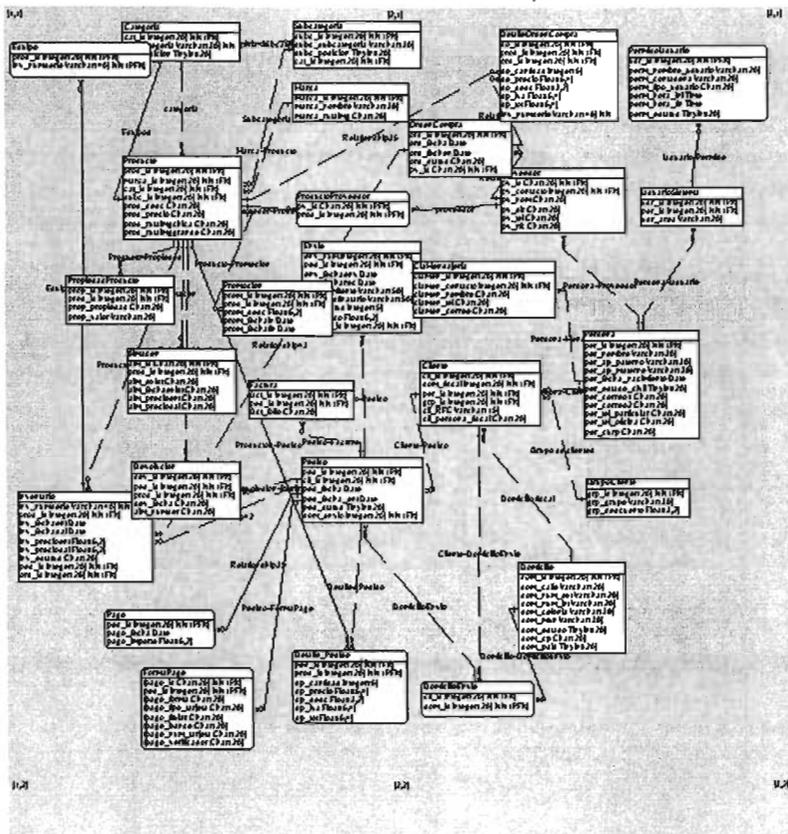


Figura 5.6.1 Estructura de la base de datos

5.7 SQL

Es un lenguaje estándar de consulta para bases de datos (Standard Query Language).

Durante este módulo se estudiaron los comandos básicos en SQL para realizar consultas, crear, eliminar y modificar tablas, creación de triggers² y algunas de las funciones más importantes para manejo de fechas y textos.

² Procedimientos que se ejecutan de manera automática cuando una estructura INSERT, UPDATE o DELETE es empleada sobre una tabla o sobre una vista; pueden estar asociados a eventos que ocurran sobre la base de datos.

Después se realizaron varios ejercicios en una base de datos MySQL instalada en el servidor web del departamento de Educación Continua de la FES Aragón. También se practicó en un equipo local con una base de datos de DB2.

MySQL³ es un gestor de bases de datos muy utilizado debido a que se distribuye bajo la licencia de código abierto y en costos, es más económico que otros gestores.

Sus desventajas, no maneja triggers ni procedimientos almacenados.

DB2 Es una base de datos desarrollada por la compañía IBM. Es muy robusta y cuenta con una suite de herramientas y soluciones personalizadas provistas por su fabricante. El costo es mayor, comparado con MySQL.

Para el caso práctico de Sumitel, se eligió usar una base de datos MySQL. Es esquema de la base de datos queda como sigue.

```
/*
Created          12/11/2004
Modified         10/06/2005
Project
Model
Company
Author
Version
Database         MySQL
*/

Create table Pedido (
    ped_id Integer(20) UNSIGNED NOT NULL AUTO_INCREMENT,
    cli_id Integer(20) NOT NULL ,
    ped_fecha Date ,
    ped_fecha_ent Date ,
    ped_status Tinyint(20) ,
    dom_envio Integer(20) NOT NULL ,
    Primary Key (ped_id)
);

Create table Detalle_Pedido (
    ped_id Integer(20) NOT NULL ,
    prod_id Integer(20) NOT NULL ,
    dp_cantidad Integer(6) ,
    dp_precio Float(6,4) ,
    dp_desc Float(3,2) ,
    dp_iva Float(6,4) ,
    dp_tot Float(6,4) ,
    Primary Key (prod_id)
);
```

³ Ver <http://www.mysql.com>

```
Create table Envio (  
    env_num Integer(20) NOT NULL ,  
    ped_id Integer(20) NOT NULL ,  
    env_fechaenv Date ,  
    env_fechaec Date ,  
    env_remitente Varchar(50) ,  
    env_destinatario Varchar(50) ,  
    env_status Integer(5) ,  
    env_peso Float(6,2) ,  
    ciamen_id Integer(20) NOT NULL ,  
    Primary Key (env_num)  
);
```

```
Create table Producto (  
    prod_id Integer(20) NOT NULL ,  
    marca_id Integer(20) NOT NULL ,  
    cat_id Integer(20) NOT NULL ,  
    subc_id Integer(20) NOT NULL ,  
    prod_desc Char(20) ,  
    prod_precio Char(20) ,  
    prod_rutaimgchica Char(20) ,  
    prod_rutaimggrande Char(20) ,  
    Primary Key (prod_id)  
);
```

```
Create table Proveedor (  
    pv_id Char(20) NOT NULL ,  
    pv_contacto Integer(20) NOT NULL ,  
    pv_nom Char(20) ,  
    pv_dir Char(20) ,  
    pv_tel Char(20) ,  
    pv_rfc Char(20) ,  
    Primary Key (pv_id)  
);
```

```
Create table Almacen (  
    alm_id Char(20) NOT NULL ,  
    prod_id Integer(20) NOT NULL ,  
    alm_exist Char(20) ,  
    alm_fechaexist Char(20) ,  
    alm_precioent Char(20) ,  
    alm_preciosal Char(20) ,  
    Primary Key (alm_id)  
);
```

```
Create table Persona (  
    per_id Integer(20) UNSIGNED NOT NULL AUTO_INCREMENT ,  
    per_nombre Varchar(30) ,  
    per_ap_paterno Varchar(30) ,  
    per_ap_materno Varchar(30) ,  
    per_fecha_nacimiento Date ,  
    per_estado_civil Tinyint(20) ,  
    per_correo1 Char(20) ,  
    per_correo2 Char(20) ,  
    per_tel_particular Char(20) ,  
    per_tel_oficina Char(20) ,  
    per_curp Char(20) ,  
    Primary Key (per_id)  
);
```

```
Create table Cliente (
    cli_id Integer(20) UNSIGNED NOT NULL AUTO_INCREMENT,
    dom_fiscal Integer(20) NOT NULL ,
    per_id Integer(20) NOT NULL ,
    grp_id Integer(20) NOT NULL ,
    cli_RFC Varchar(15) ,
    cli_persona_fiscal Char(20) ,
    Primary Key (cli_id)
);

Create table GrupoCliente (
    grp_id Integer(20) UNSIGNED NOT NULL AUTO_INCREMENT,
    grp_grupo Varchar(30) ,
    grp_descuento Float(3,2) ,
    Primary Key (grp_id)
);

Create table Categoria (
    cat_id Integer(20) UNSIGNED NOT NULL AUTO_INCREMENT,
    cat_categoria Varchar(30) NOT NULL ,
    cat_posicion Tinyint(20) ,
    Primary Key (cat_id)
);

Create table UsuarioSistema (
    usr_id Integer(20) UNSIGNED NOT NULL AUTO_INCREMENT,
    per_id Integer(20) NOT NULL ,
    usr_area Varchar(20) ,
    Primary Key (usr_id)
);

Create table ProductoProveedor (
    pv_id Char(20) NOT NULL ,
    prod_id Integer(20) NOT NULL ,
    Primary Key (pv_id,prod_id)
);

Create table Subcategoria (
    subc_id Integer(20) UNSIGNED NOT NULL AUTO_INCREMENT,
    subc_subcategoria Varchar(30) ,
    subc_posicion Tinyint(20) ,
    cat_id Integer(20) NOT NULL ,
    Primary Key (subc_id)
);

Create table Inventario (
    inv_numserie Varchar(40) BINARY NOT NULL ,
    prod_id Integer(20) NOT NULL ,
    inv_fechaent Date ,
    inv_fechaasal Date ,
    inv_precioent Float(6,2) ,
    inv_preciosal Float(6,2) ,
    inv_estatus Char(20) ,
    ped_id Integer(20) NOT NULL ,
    ord_id Integer(20) NOT NULL ,
    Primary Key (inv_numserie)
);

Create table Equipo (
    prod_id Integer(20) NOT NULL ,
    inv_numserie Varchar(40) NOT NULL ,
    Primary Key (prod_id,inv_numserie));
```

```
Create table Devolucion (  
    dev_id Integer(20) NOT NULL AUTO_INCREMENT,  
    ped_id Integer(20) NOT NULL ,  
    prod_id Integer(20) NOT NULL ,  
    dev_fecha Char(20) ,  
    alm_numser Char(20) ,  
    Primary Key (dev_id)  
);
```

```
Create table PermisoUsuario (  
    usr_id Integer(20) NOT NULL ,  
    perm_nombre_usuario Varchar(20) ,  
    perm_contasena Varchar(20) ,  
    perm_tipo_usuario Char(20) ,  
    perm_hora_ini Time ,  
    perm_hora_fin Time ,  
    perm_estatus Tinyint(20) ,  
    Primary Key (usr_id)  
);
```

```
Create table CiaMensajeria (  
    ciamen_id Integer(20) NOT NULL AUTO_INCREMENT,  
    ciamen_contacto Integer(20) NOT NULL ,  
    ciamen_nombre Char(20) ,  
    ciamen_tel Char(20) ,  
    ciamen_correo Char(20) ,  
    Primary Key (ciamen_id)  
);
```

```
Create table Marca (  
    marca_id Integer(20) NOT NULL AUTO_INCREMENT,  
    marca_nombre Varchar(30) ,  
    marca_rutaimg Char(20) ,  
    Primary Key (marca_id)  
);
```

```
Create table Domicilio (  
    dom_id Integer(20) NOT NULL AUTO_INCREMENT,  
    dom_calle Varchar(20) ,  
    dom_num_ext Varchar(20) ,  
    dom_num_int Varchar(20) ,  
    dom_colonia Varchar(30) ,  
    dom_mun Varchar(30) ,  
    dom_estado Tinyint(20) ,  
    dom_cp Char(20) ,  
    dom_pais Tinyint(20) ,  
    Primary Key (dom_id)  
);
```

```
Create table DomicilioEnvio (  
    cli_id Integer(20) NOT NULL ,  
    dom_id Integer(20) NOT NULL ,  
    Primary Key (dom_id)  
);
```

```
Create table Promocion (  
    prom_id Integer(20) NOT NULL ,  
    prod_id Integer(20) NOT NULL ,  
    prom_desc Float(6,2) ,  
    prom_fechain Date ,  
    prom_fechafin Date ,  
    Primary Key (prom_id);
```

```
Create table PropiedadProducto (
    prop_id Integer(20) NOT NULL ,
    prod_id Integer(20) NOT NULL ,
    prop_propiedad Char(20) ,
    prop_valor Varchar(20) ,
    Primary Key (prop_id)
);

Create table OrdenCompra (
    ord_id Integer(20) UNSIGNED NOT NULL AUTO_INCREMENT,
    ord_fecha Date ,
    ord_fechen Date ,
    ord_status Char(20) ,
    pv_id Char(20) NOT NULL ,
    Primary Key (ord_id)
);

Create table DetalleOrdenCompra (
    do_id Integer(20) NOT NULL AUTO_INCREMENT,
    prod_id Integer(20) NOT NULL ,
    ord_id Integer(20) NOT NULL ,
    do_cantidad Integer(6) ,
    do_precio Float(6,4) ,
    do_desc Float(3,2) ,
    dp_iva Float(6,4) ,
    dp_tot Float(6,4) ,
    inv_numserie Varchar(40) NOT NULL ,
    Primary Key (do_id)
);

Create table FormaPago (
    fpago_id Char(20) NOT NULL ,
    ped_id Integer(20) NOT NULL ,
    fpago_forma Char(20) ,
    fpago_tipo_tarjeta Char(20) ,
    fpago_titular Char(20) ,
    fpago_banco Char(20) ,
    fpago_num_tarjeta Char(20) ,
    fpago_verificador Char(20) ,
    Primary Key (fpago_id,ped_id)
);

Create table Factura (
    fact_id Integer(20) NOT NULL AUTO_INCREMENT,
    ped_id Integer(20) NOT NULL ,
    fact_folio Char(20) ,
    Primary Key (fact_id)
);

Create table Pago (
    ped_id Integer(20) NOT NULL ,
    pago_fecha Date ,
    pago_importe Float(6,2) ,
    Primary Key (ped_id)
);
```

5.8 Seguridad

Estando nuestro sistema en una red pública como Internet, siempre será vulnerable a ataques. Uno de los ataques más comunes es la inyección de código SQL en el query string que envía el explorador, cuando se envían datos con el método get.

Algunas recomendaciones para evitar esto son:

- Filtrar la información en el lado del servidor y no del cliente.
- Limitar los permisos de ejecución de sentencias en el gestor de bases de datos.
- Validar la longitud de los datos que ingresa el usuario.
- Remover todos los datos reconocidos como dañinos y permitir sólo caracteres válidos.
- Preferentemente, tener un servidor para servir las aplicaciones y otro para bases de datos.
- Configurar los reportes de error para no mostrar la estructura de las tablas.
- En nuestro caso al usar JDBC, utilizar Prepare Statement.
- Preferir transacciones lentas y seguras.
- Evitar el uso de SELECT *.
- Dar alias a las tablas y los campos.

Capítulo 6

Reglas de negocio aplicadas a un sitio de comercio electrónico.

6.1 Reglas de negocio

Todas las empresas tienen definidas restricciones, normas, reglas y validaciones para la información que manejan durante sus procesos; este compendio de ideas se conoce como reglas de negocio.

Cuando se diseña un sistema para determinada empresa, es crucial conocer sus reglas de negocio para que se cumpla en un alto porcentaje con el objetivo principal; si el sistema no cumple con algunas de las reglas es muy probable que ni siquiera sea utilizado.

No es una tarea sencilla, pero se debe cumplir con la mayoría de estas reglas.

El modelo de tres capas tiene una capa de reglas de negocio y es aquí en donde se lleva a cabo toda la lógica que permitirá cubrir estas necesidades.

Para poder implementar el modelo de tres capas en un sistema web se requiere un servidor de aplicaciones, que es un programa que reside en un servidor que da servicio a una red distribuida.

Para el caso de estudio que se está presentando no es suficiente un contenedor web como Jakarta Tomcat¹, por lo que se usará Apache Struts² que es un entorno de desarrollo para aplicaciones en Java que trabaja bajo el paradigma Modelo-Vista-Controlador (MVC)³ y hará las veces de servidor de aplicaciones, aunque no en sentido estricto. Más adelante se explicará porqué.

Aprovechando las ventajas de trabajar con componentes se instalarán los java beans de Ibatis⁴ para realizar toda la interactividad con bases de datos.

¹ <http://jakarta.apache.org/tomcat/>

² <http://struts.apache.org/>

³ Es un patrón de diseño de software que separa el modelo de datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos de forma que las modificaciones al componente de la vista pueden ser hechas con un mínimo impacto en el componente del modelo de datos.

⁴ <http://ibatis.apache.org/>

6.2 Definición

Las reglas de negocio son todos aquellos lineamientos que definen el funcionamiento de una empresa, sin ellas no habría orden.

En todas las empresas hay flujos de trabajo que, por lo general, no son lineales; esto hace que sea muy difícil cumplir al 100% con las reglas de negocio.

Pero si un sistema que no cumple con las reglas de negocio es un sistema inservible porque simplemente no será usado.

Reglas de modelo de datos

Definen las características básicas de los datos. Por ejemplo: "Los estatus de los pedidos estarán representados por números enteros sin signo."

Reglas de relación

Son aquellas que definen la manera en que estarán relacionados los datos y tienen que ver directamente con la integridad de los mismos.

Reglas de derivación

Indican la manera en que los datos existentes se relacionan para generar nuevos datos. Como ejemplo podemos tomar el importe total de un pedido que siempre podrá ser calculado a partir del importe de las partidas que lo conforman para evitar almacenarlo en la base de datos.

Reglas de restricción

Establecen las restricciones sobre los datos más allá de las definidas por las reglas del modelo de datos. Por ejemplo: una regla de modelado de datos es "La edad de un usuario debe ser un número entero sin signo", la regla de restricción es: "Los usuarios deben ser mayores de 18 años y menores de 70".

Reglas de flujo

Las reglas de negocio que definen y delimitan del flujo de trabajo, son el resultado de un estudio exhaustivo de las operaciones de la empresa.

Sintetizando, se puede decir que el propósito de una regla de negocio es capturar lo que está o no está permitido en una determinada empresa.⁵

6.3 Arquitecturas

Como anteriormente se había visto existen varias arquitecturas para el desarrollo de sistemas, unas de las más conocidas la de dos y tres capas.

En la arquitectura de dos capas conocida también como cliente-servidor existe una capa de presentación y otra de reglas de negocio y datos, comunicadas por una aplicación intermedia comúnmente llamada middleware.

En la arquitectura de tres capas la capa de negocios y la de datos conforman capas separadas.

Una aplicación para el Web desarrollada bajo la arquitectura de tres capas está integrada como muestra la siguiente figura:

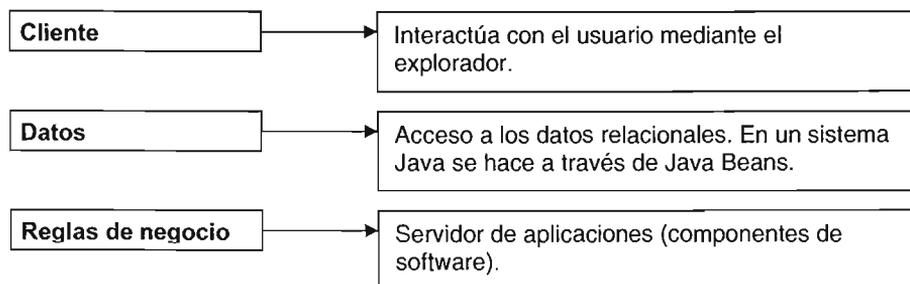


Figura 6.3.1. Aplicación para el Web desarrollada bajo la arquitectura de tres capas

6.3.1. Servidor de aplicaciones

Un servidor de aplicaciones es un software especial que maneja aplicaciones de misión crítica mediante un esquema transaccional. Se entiende por transacción un conjunto de instrucciones ejecutadas en orden y cuyo cumplimiento exitoso indica si la transacción ha sido concluida.

⁵ El valor de las reglas de negocio en los procesos. <http://www.ecm-spain.com/interior.asp?IdItem=5551>

Para un sistema desarrollado en java (como el presentado en este trabajo) el servidor de aplicaciones está compuesto por un Web container y un EJB container; la distinción entre estos contenedores no es muy clara. Actualmente existen en el mercado varios servidores de aplicaciones en los que ambos contenedores parecen ser uno sólo debido a su alta integración ("Fully J2EE Compliant"), por ejemplo, WebLogic, Websphere y Oracle 9i Application Server.

Por otro lado, existen productos que son sólo EJB containers, JBoss es uno de ellos, y otros que únicamente son Web containers como es el caso de Jakarta Tomcat.

El web container es un servidor que nativamente comprende:

- El protocolo http
- Java Server pages
- Servlets
- Java beans

El EJB container sirve aplicaciones desarrolladas con Enterprise Java Beans mejor conocidos como EJB's.

6.3.2. *Jakarta Struts*

Es un framework ⁶de código abierto para desarrollo de sistemas web en Java, no es estrictamente un servidor de aplicaciones. Sin embargo es recomendable para desarrollar aplicaciones web que no requieran transacciones críticas.

Struts trabaja bajo el paradigma Modelo-Vista-Controlador (MVC). La separación en estos tres niveles lo hace independiente de la tecnología a utilizar y hace más rápido el desarrollo.

El controlador de vistas sabe quien puede recibir peticiones y quien debe devolver los resultados a través de archivos de configuración en lenguaje xml, esto permite que las aplicaciones generadas sean estándar. Los pasos serán siempre los mismos ya que se trata de un patrón.

⁶ Conjunto de clases especializadas

Para el caso ejemplificado en este trabajo se optó por utilizar Jakarta Tomcat y Struts, básicamente porque ambas son aplicaciones de libre distribución y porque son las recomendadas para iniciarse en el desarrollo de sistemas web con Java.

6.3.3. *iBatis*

*iBatis*⁷ es un framework para acceso de bases de datos con Java y .NET, estrictamente se trata de una herramienta ORM (Object Relational Mapping). Un ORM realiza un mapeo entre objetos (Java Beans) y una base de datos relacional a través de un archivo de configuración (mapa). *iBatis* es una herramienta que proporciona un cierto grado de libertad al desarrollador ya que no está comprometida con un gestor de bases de datos en específico.

6.4 Logística del Sitio Web

El sistema en Internet promueve la colaboración de servicios a través de la interacción de objetos en línea.

Bajo la arquitectura de Struts, tenemos un objeto (Java Bean) encargado de crear nuevos objetos del modelo, se conoce como *ActionObject*, dependiendo de la petición que reciba.

El *ActionObject* recibe un *HttpRequest*⁸ y lo convierte a un *HttpServletRequest*⁹. Busca la acción correspondiente en el modelo, obtiene una respuesta y después investiga cómo y a quien debe entregar los resultados obtenidos. En el siguiente diagrama se puede ver con más claridad la forma en que actúa struts.

⁷ <http://www.ibatis.com/about.html>

⁸ *HttpRequest*

⁹ *HttpServletRequest*

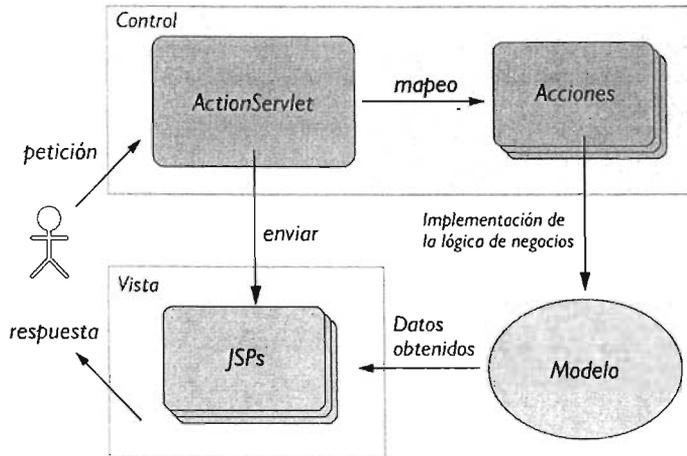


Figura 6.4 Implementación de una aplicación web con Struts

Para desarrollar una aplicación de ejemplo fue necesario instalar el plug-in de Tomcat para Eclipse, esta herramienta permite crear un proyecto web de Tomcat y compilarlo cada vez que se guarda.

Es necesario conocer la estructura de un proyecto Web para poder configurarlo:

HOME-PROYECTO

/WEB-INF

lib

classes

src

*.xml

*.tjs

/work

Dentro del directorio lib estarán todas las librerías externas necesarias para que funcione la aplicación.

Los Java Beans se encuentran en el directorio classes.

El archivo de configuración de Struts se encuentra en WEB-INF y se llama struts-config.xml y es aquí en donde se definen los Java Beans y se definen las peticiones y las repuestas.

Conclusiones

El modelado cuidadoso de un sistema informático aumenta las posibilidades de éxito del mismo. Los desarrolladores pueden asegurarse que la funcionalidad será la correcta y así, la cobertura de los requerimientos del cliente más amplia, de manera que, factores como escalabilidad, seguridad y robustez están casi asegurados.

La seguridad en los sistemas web es un punto crucial a tomar en cuenta al momento de analizar y diseñar la estructura lógica y física del mismo. Con base a la experiencia adquirida durante este diplomado se llegó a la conclusión de que un modelo que permita aislar los datos y la lógica de nuestro sistema de la presentación e interactividad con usuario, es el ideal para minimizar el riesgo de potenciales corrupciones al sistema.

Otro punto muy importante es el diseño de una base de datos robusta, que cubra todas las posibles grietas que pudieran permitir una acción maligna contra los datos del sistema.

Por el lado de la interacción con el usuario, los sistemas basados en el web están en desventaja con las aplicaciones de escritorio, ya que se depende mucho del ambiente en que se este utilizando la aplicación lo cual depende directamente de el entorno de cada usuario (velocidad de conexión, computadora, navegador, etc.).

Encontrar el canal adecuado para interactuar con el usuario es una tarea que requiere de un análisis a fondo, no sólo se trata de cumplir con los requerimientos del usuario, sino de lograr que la experiencia con el sistema se vuelva algo natural.

Bibliografía

LARMAN, Craig. *Applying UML and Patterns, An Introduction to Object-Oriented Analysis and Design*. 2ª ed. New Jersey: Prentice Hall, 2001.

SCHNEIDER, Geri; WINTERS, Jason P. *Applying Use Cases. A Practical Guide*. 2ª ed. Massachusetts: Addison Wesley, 2001.

GAMMA, Erich; et. al. *Design Patterns. Elements of Reusable Object-Oriented Software*. Massachusetts: Addison Wesley, 1995.

LAZAR, Jonathan. *<User-Centered Web Development>*. 1ª ed. U.S.A.: Jones and Bartlett Computer Science, 2001. p. 108-122, 156-194

RICHARDSON, Eric C., *Programming Web Server Applications*. 1ª ed. U.S.A.: Prima Publishing, 1999. p. 4-14.

ROWE, Jeff. *Building Internet Database Servers with CGI*. 1ª ed. U.S.A.: New Riders Publishing, 1996. p. 70-95, 98-130, 139-173.

MAGID, Jonathan, et. al. *The Web Server Book*. 1ª ed. E.U.A.: Ventana Press, 1995. p. 1-17, 26-31, 71-100, 105-146.

NELL, Dale, et. al. *Object-oriented data structures using Java*. 1ª ed. U.S.A.: Jones and Bartlett, c2002. p. 2-6, 9-14, 70-118.

CEBALLOS, Francisco Javier. *Java 2™ Interfaces gráficas y aplicaciones para Internet*. 1ª ed. Madrid : Ra-Ma, c2005. p. 233-244, 246-271, 337-392, 413-421, 527-538.

Índice

Objetivos..... I

Introducción..... II

Capítulo 1. Generalidades del desarrollo en web, orientadas a un sitio de comercio electrónico.

1.1 Antecedentes.....	1
1.2 Modelo de 3 capas.....	2
1.3 Hardware.....	3
1.3.1 El servidor Web.....	3
1.3.2 Estaciones de trabajo para desarrolladores.....	3
1.4 Software.....	4
1.4.1 Sistema operativo.....	4
1.4.2 Servidor Web Apache.....	5
1.4.3 Secure Shell.....	5
1.4.4 JDK (Java Developer Kit).....	5
1.4.5 Servidor de aplicaciones Tomcat.....	6
1.4.6 Manejador de bases de datos.....	6
1.5 Propuesta del proyecto.....	6
1.5.1 Generalidades.....	7
1.5.2 Facilidad de uso y exploración.....	8
1.5.3 Rápido rendimiento.....	9
1.5.4 Reducción del tiempo de respuesta.....	9
1.5.5 Compras anónimas (Exploración).....	10
1.5.6 Perfil de usuario.....	10
1.5.7 Competitividad gracias a la facilidad de uso.....	11
1.6 Características particulares del proyecto.....	11
1.6.1 ¿Qué problema resuelve este proyecto?.....	11
1.6.2 Objetivo del proyecto.....	11
1.6.3 ¿Cuál es el alcance de este proyecto?.....	12
1.6.4 Objetivos acordados.....	12
1.6.5 Ambiente.....	12
1.6.6 Back office.....	13
1.6.7 Actores y funciones.....	13

1.6.8 Panorama actual.....	15
1.6.9 Panorama automatizado.....	16
1.6.10 Tareas nuevas.....	17
1.6.11 Expectativas.....	18
1.7 Alcance.....	19
1.8 Entregables.....	20
1.9 Riesgos y beneficios.....	20
1.9.1 Riesgos.....	20
1.9.2 Beneficios.....	20
1.10 Propuesta técnica.....	20

Capítulo 2. Análisis UML para sitio web.

2.1 Generalidades.....	27
2.2 Análisis.....	28
2.3 Ciclo de vida de un sistema.....	28
2.3.1 Encuesta.....	28
2.3.2 Hardware y software.....	29
2.3.3 Factibilidad.....	29
2.3.4 Estimación.....	29
2.3.5 Análisis de Costo-Beneficio.....	29
2.4 Diseño.....	29
2.4.1 Diccionario de datos.....	31
2.4.2 Diagrama entidad relación.....	40
2.5 Modelado.....	41
2.5.1 Pasos para el modelado orientado a objetos.....	41
2.5.2 UML.....	43
2.5.3 Listado de casos de uso.....	45
2.5.4 Listado de casos de uso y actores.....	46
2.5.5 Definición de los casos de uso.....	47
2.6 Mejores prácticas aplicadas al diseño de sistemas.....	71

Capítulo 3. Java como herramienta de desarrollo, un enfoque orientado a objetos.

3.1 Características generales de Java.....	72
3.2 La plataforma Java.....	73
3.2.1 Máquina virtual.....	73
3.2.2 API.....	73
3.3 Tipos de aplicaciones.....	75
3.4 JDK (Java Developer Kit).....	76
3.5 Entorno de desarrollo Eclipse.....	76
3.6 Estándar de codificación.....	77
3.7 Calidad y procesos de software.....	77

Capítulo 4. Interfaz de usuario y sus aplicaciones.

4.1 La importancia de una buena interfaz.....	79
4.2 Niveles de usuario.....	80
4.3 Mapa de navegación.....	80
4.4 Normatividad.....	81
4.5 Bosquejo de diseño.....	81
4.6 Prototipo.....	81
4.7 Evaluación.....	85
4.7.1 Usabilidad.....	85
4.8 Herramientas de diseño.....	87
4.8.1 Html.....	87
4.8.2 Macromedia™ Dreamweaver™.....	87
4.8.3 Javascript.....	87
4.9 Contenido dinámico.....	88

Capítulo 5. Fundamentos y aplicaciones de bases de datos.

5.1 La capa de datos.....	89
5.2 Sistema de gestión de bases de datos.....	89
5.3 Usuarios en el SGBD.....	90
5.4 Funciones de un SGDB.....	90
5.5 Características deseables de una base de datos.....	91
5.6 Modelo relacional de bases de datos.....	91
5.7 SQL.....	92
5.8 Seguridad.....	98

Capítulo 6. Reglas de negocio aplicadas a un sitio de comercio electrónico.

6.1 Reglas de negocio.	99
6.2 Definición.....	100
6.3 Arquitecturas.....	101
6.3.1 Servidor de aplicaciones.....	101
6.3.2 Jakarta Struts.....	102
6.3.3 iBatis.....	103
6.4 Logística del Sitio Web.....	103
Conclusiones.....	106
Bibliografía.....	107
Índice de imágenes	
Figura 1.2.1 Modelo de tres capas.....	2
Figura 1.10.1 Logística del sitio web.....	23
Figura 2.4.1 Diagrama entidad relación.....	40
Figura 2.5.4 Diagramas de caso de uso.....	70
Figura 3.2.1 Ejecución de un programa a través de la máquina virtual.....	74
Figura 4.6.1. Página principal de la tienda virtual.....	82
Figura 4.6.2. Detalles de un producto.....	83
Figura 4.6.3. Vista del carrito de comprar con productos.....	83
Figura 4.6.4. Vista de un pedido antes de confirmarse.....	84
Figura 4.6.5. Listado de pedidos para un cliente.....	84
Figura 4.6.6. Primera pantalla para registro de un cliente.....	85
Figura 6.3.1 Aplicación para el Web desarrollada bajo la arquitectura de tres capas.....	101
Figura 6.4 Implementación de una aplicación web con Struts.....	104