

ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO

ESCUELA DE POSTGRADO

MAESTRIA EN INFORMATICA APLICADA

TESIS DE GRADO

**“DIRECTRICES DE INTEROPERABILIDAD DE UNA BASE DE DATOS DISTRIBUIDA
PARA LA OPTIMIZACION DE SISTEMAS DE COMERCIO ELECTRÓNICO”**

MAESTRANTE

ING. MARCO VINICIO YAMBAY DIAZ

TUTOR

ING. MS. WILSON BALDEON

Remitido a la Escuela de Postgrado de la Escuela Superior Politécnica de Chimborazo

Como requisito final previo a la obtención del título de

MASTER EN INFORMATICA APLICADA

Riobamba, 2004

CERTIFICACIÓN DEL ASESOR

Certifico que el presente trabajo de investigación sobre “DIRECTRICES DE INTEROPERABILIDAD DE UNA BASE DE DATOS DISTRIBUIDA PARA LA OPTIMIZACION DE SISTEMAS DE COMERCIO ELECTRÓNICO” fue realizado por el Ing. Marco Vinicio Yambay Díaz bajo mi asesoramiento, por lo que autorizo su presentación.

Atentamente,

Ing. Ms. Wilson Baldeón

TUTOR DE TESIS

DERECHOS DE AUTORIA

Yo, Marco Vinicio Yambay Díaz, soy responsable de las ideas expuestas en el presente trabajo de investigación y, los derechos de autoría pertenecen a la Escuela Superior Politécnica de Chimborazo.

DEDICATORIA

A mi familia, quienes con sus consejos
me impulsan a seguir alcanzando nuevas metas.

AGRADECIMIENTO

A mi familia por su apoyo para cumplir con este objetivo,
A COMPUTEC por su colaboración en la realización de este trabajo,
A la ESPOCH por brindar la oportunidad de una superación científica.

INDICE

	Pág.
PORTADA	
DERECHOS DE AUTORIA	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
INDICE	vi
LISTA DE FIGURAS	xiii
LISTA DE TABLAS	xv
RESUMEN	xvi
SUMMARY	xvii
INTRODUCCIÓN	18
PLANTEAMIENTO DEL PROBLEMA	19
OBJETIVOS	19
JUSTIFICACION	20
CAPITULO I	
SISTEMAS DE BASES DE DATOS DISTRIBUIDAS EN INTERNET	22
1.1 BASE DE DATOS DISTRIBUIDA (DDB)	22
1.2 ARQUITECTURA DE UNA DDB EN INTERNET	23
1.3 METODOLOGÍAS DE DISEÑO	23
1.3.1 METODO TOP-DOWN O ENFOQUE DESCENDENTE	24

1.3.1.1 DISEÑO DE DISTRIBUCIÓN	27
1.3.1.2 MANEJO DE LA FRAGMENTACIÓN	27
1.3.1.3 TIPOS DE FRAGMENTACION	28
1.3.1.3.1 FRAGMENTACIÓN HORIZONTAL	28
1.3.1.3.2 FRAGMENTACIÓN VERICAL	29
1.3.1.3.3 FRAGMENTACIÓN MIXTA O HIBRIDA	29
1.3.1.4 ASIGNACIÓN O COLOCACIÓN	30
1.3.1.4.1 ASIGNACIÓN REDUNDANTE	31
1.3.1.4.2 ASIGNACIÓN NO REDUNDANTE	31
1.3.2 METODO BOTOM-UP O ENFOQUE ASCENDENTE	31
1.4 PROGRAMAS CGI COMO INTERFAZ PARA ACCEDER A BASES DE DATOS DISTRIBUIDA	32
1.5 PROCESO DE EXPLORACIÓN EN UNA DDB CON CGI	35
1.6 MANEJO DISTRIBUIDO DE TRANSACCIONES	37
1.7 PROTOCOLOS DE CONTROL DE CONCURRENCIA	37
CAPITULO II	
SISTEMAS DE COMERCIO ELECTRONICO	39
2.1 MODELO FUNCIONAL CLIENTE - SERVIDOR DE INTERNET	39
2.1.1 PROGRAMA CLIENTE (NAVEGADOR WEB)	40
2.1.2 PROGRAMA SERVIDOR (SERVIDOR WEB)	40
2.2 INFRAESTRUCTURA PARA EL E-COMERCE	41
2.3 CONSIDERACIONES EN EL DESARROLLO DE APLICACIONES E-COMMERCE	42
2.3.1 ANÁLISIS PRELIMINAR	42
2.3.2 REGLAS COMERCIALES	43
2.4 ARQUITECTURA DEL SISTEMA DE COMERCIO ELECTRONICO	43

2.4.1 ROLES Y FUNCIONES	44
2.4.1.1 ROLES	44
2.4.1.2 ROLES DEL COMPRADOR O CLIENTE	45
2.4.1.3 ROLES DEL VENDEDOR O PROVEEDOR	46
2.4.2 COMPONENTES FUNCIONALES	47
2.4.2.1 COMPONENTES DEL COMPRADOR O CLIENTE	48
2.4.2.2 COMPONENTES DEL VENDEDOR O PROVEEDOR	48
2.5 PROCESO TRANSACCIONAL DEL CONTENIDO CON CGI	49
2.6 TRANSFERENCIA DE PARAMETROS ENTRE EL WEB Y UN PROGRAMA CGI	52
2.7 ELABORACION DE UN MODELO CONFIABLE	52
2.8 SEGURIDADES EN LA INTERCONEXION DE LA WEB CON LA BD	53
2.8.1 PROTOCOLOS DE SEGURIDAD	54
2.8.2 PERMISOS DE ACCESO	54
2.8.3 CONSIDERACIONES PARA LOS EJECUTABLES	55
CAPITULO III	
MARCO HIPOTETICO	56
3.1. HIPOTESIS	56
3.2 OPERACIONALIZACION DE VARIABLES	57
CAPITULO IV	
ANALISIS DE LAS DIRECTRICES DE UNA BASE DE DATOS DISTRIBUIDA PARA LA OPTIMIZACION DE SISTEMAS DE COMERCIO ELECTRONICO	58
4.1 ESQUEMA DEL PROCESO DE DISTRIBUCION	58
4.2 REQUERIMIENTOS DE INFORMACION	59
4.3 PROCESO DE FRAGMENTACIÓN	60

4.3.1 FRAGMENTACION HORIZONTAL	60
4.3.1.1 FRAGMENTACION HORIZONTAL PRIMARIA	61
4.3.1.1.1 COMPLETITUD Y MINIMALIDAD DE LOS PREDICADOS SIMPLES	62
4.3.1.1.2 GENERACION DE PREDICADOS MINITERMINO	64
4.3.1.1.3 DEFINICION DE IMPLICACIONES	65
4.3.1.1.4 ALGORITMO PARA LA FRAGMENTACION HORIZONTAL PRIMARIA	68
4.3.1.2 FRAGMENTACION HORIZONTAL DERIVADA	68
4.3.2 FRAGMENTACION VERTICAL	70
4.3.2.1 DEFINICION DE USO DE ATRIBUTOS	71
4.3.2.2 FRECUENCIA DE ACCESO POR SITIO	71
4.3.2.3 DEFINICION DE AFINIDAD ENTRE ATRIBUTOS	72
4.3.2.4 AGRUPACION ENTRE ATRIBUTOS	73
4.3.2.5 ALGORITMO DE AGRUPACION	73
4.3.2.6 ALGORITMO DE FRAGMENTACION VERTICAL	74
4.3.3 FRAGMENTACION MIXTA	76
4.4 PROCESO DE ASIGNACION	76
4.4.1 INFORMACION NECESARIA	76
4.4.2 OPTIMIZACION DE LOS COSTOS DE LA ASIGNACION	78
4.4.3 CONSIDERACION EN LAS ALTERNATIVAS DE ASIGNACION	79
4.4.3.1 MAYOR DISPONIBILIDAD	79
4.4.3.2 INCREMENTO DEL PARALELISMO	79
4.4.3.3 MAYOR TIEMPO EXTRA EN ACTUALIZACIONES	79
4.4.3.4 RECUPERACION	80
4.5 FACTORES ADICIONALES A CONSIDERAR EN EL RENDIMIENTO DE UN BBD	80

4.5.1 AFINAMIENTO DEL SISTEMA OPERATIVO	80
4.5.2 OPTIMIZACION EN LA TECNOLOGÍA DE ALMACENAMIENTO	82
4.5.3 ALMACENAMIENTO PRIMARIO	82
4.5.4 RESPALDO	83
4.5.5 ELECCIÓN DE LA TECNOLOGÍA RAID	84
4.6 OPTIMIZACION DEL PROCESAMIENTO DE CONSULTAS	85
4.7 AFINAMIENTO DEL CONECTOR DE BASE DE DATOS INTERNET (IDC)	85
4.8 OPTIMIZACION DE CONSULTAS	86
4.8.1 POR AGRUPACIÓN DE CONSULTAS POR LOTES	86
4.8.2 POR AGRUPACIÓN DE CONSULTAS MULTIPLES	87
4.9 CONSIDERACIONES EN LOS ENTORNOS DISTRIBUIDOS	88
4.9.1 PROBLEMAS DE LOS ENTORNOS DISTRIBUIDOS	88
4.9.2 SUBSISTEMAS EN LOS ENTORNOS DISTRIBUIDOS	89
4.10 OPTIMIZACION DE LA INTRANET	89
CAPITULO V	
PROPUESTA PARA LA IMPLEMENTACION DE DIRECTRICES	91
5.1 INTRODUCCION	91
5.2 OBJETIVOS	91
5.3 REQUERIMIENTOS PARA IMPLEMENTACION	92
5.3.1 SOFTWARE	92
5.3.2 HARDWARE	92
5.4 METODO PARA LA REALIZACION	93
5.4.1 ESCENARIO DEL SISTEMA DE COMERCIO ELECTRÓNICO	93
5.4.1.1 LA BASE DE DATOS	94
5.4.1.2 LA APLICACIÓN DE COMERCIO ELECTRÓNICO	94

5.4.1.3 CONFIGURACION DEL SERVIDOR WEB Y DE BASE DATOS	101
5.4.1.4 CONFIGURACIÓN DE LOS ORÍGENES DE DATOS ODBC	104
5.4.2 POBLACION Y MUESTRA	104
5.4.3 METODOS E INSTRUMENTOS DE RECOLECCION DE DATOS	105
5.4.4 TECNICAS DE PROCESAMIENTO Y ANALISIS DE DATOS	105
5.4.4.1 DEFINICION DE METRICAS	105
5.4.4.2 EVALUACIÓN DE METRICAS	106
5.5 PROCESAMIENTO Y ANALISIS DE LAS ESTADISTICAS DE RENDIMIENTO	109
5.6 PRUEBA DE HIPOTESIS	114
CAPITULO VI	
PROPUESTA DE DIRECTRICES EJECUTIVAS	115
6.1 DIRECTRICES DE CARÁCTER GENERAL	115
6.2 DIRECTRICES DE CARÁCTER ESPECIFICO	116
CONCLUSIONES Y RECOMENDACIONES	120
CONCLUSIONES	120
RECOMENDACIONES	121
BIBLIOGRAFÍA	123
BIBLIOGRAFÍA GENERAL	123
BIBLIOGRAFIA ESPECIALIZADA	123
DIRECCIONES INTERNET	124
ANEXOS	

I: ANTECEDENTES DE LOS SISTEMAS DE COMERCIO ELECTRÓNICO	126
II: REGLAS COMERCIALES DE COMERCIO ELECTRONICO	136
III: CODIGO DE LA APLICACIÓN WEB DE COMERCIO ELECTRONICO	145
IV: CONFIGURACIONES AVANZADAS DE LA CONEXIÓN A BASE DE DATOS	170

LISTA DE FIGURAS

	Pág.
Figura 1.- Esquema de Diseño Conceptual	25
Figura 2.- Esquema de Diseño Lógico	25
Figura 3.- Esquema de Diseño Físico	25
Figura 4.- Diseño de distribución - asignación	26
Figura 5.- Diseño físico - fragmentación	26
Figura 6.- Fragmentación - asignación	27
Figura. 7.- Fragmentación horizontal	28
Figura. 8.- Fragmentación vertical	29
Figura. 9.- Esquema fragmentación híbrida	30
Figura 10.- Esquema servidor HTTP, BD, CGI	34
Figura 11.- Esquema exploración de BD con un CGI	35
Figura 12.- CGI en el proceso transaccional	49
Figura 13.- Operaciones con tarjeta de crédito.	51
Figura 14.- Esquema enfoque en dos fases	59
Figura 15.- Esquema alternativas de fragmentación.	60
Figura 16.- Esquema fragmentación horizontal- subrelaciones	61
Figura 17.- Esquema de algoritmo de completitud y minimalidad	63
Figura 18.- Esquema semi-join de una relación	69
Figura 19.- Esquema fragmentación vertical - subrelación	70
Figura 20.- Diseño de la base de datos	94
Figura 21.- Diagrama del prototipo de la aplicación de comercio electrónico	95
Figura 22.- Prototipo página de inicio	96
Figura 23.- Prototipo página de registro	97

Figura 24.- Prototipo página de pedidos	98
Figura 25.- Prototipo segmentación del pedido	99
Figura 26.- Prototipo página de pedido cantidad	99
Figura 27.- Prototipo página consolidación del pedido	100
Figura 28.- Prototipo página forma de pago	101
Figura 29.- Configuración del servidor WEB	102
Figura 30.- Autenticación para SQL SERVER	102
Figura 31.- Creación de fuente de datos	103
Figura 32.- Configuración del servicio WEB	103
Figura 33.- Transacciones de pedidos en servidor	109
Figura 34.- Transacciones de pedidos en cliente	109
Figura 35.- Transacciones clientes en servidor	110
Figura 36.- Transacciones clientes en clientes	110
Figura 37.- Transacciones de clientes en servidor fragcionamiento vertical	111
Figura 38.- Transacciones de clientes en cliente fragcionamiento vertical	111
Figura 39.- Transacciones de clientes en servidor fragcionamiento horizontal	112
Figura 40.- Transacciones de clientes en cliente fragcionamiento horizontal	112

LISTA DE TABLAS

	Pág.
Tabla 1. Operacionalizacion de variables	57
Tabla 2. Predicados miniterminos	66
Tabla 3. Predicados miniterminos- implicaciones	67
Tabla 4. Matriz de Uso de Atributos	71
Tabla 5. Matriz de Acceso por Sitios	72
Tabla 6- Matriz de Afinidad entre Atributos	73
Tabla 7- Matriz agrupada de Atributos	74
Tabla 8- Matriz accesadas en conjunto	74

RESUMEN

En un sistema de comercio electrónico, uno de los puntos neurálgicos en su rendimiento es el diseño de la base de datos, por lo que el proceso de distribución debe ser abordado con criterios de eficiencia y optimalidad. Esta investigación, tomando en cuenta factores cuantitativos inherentes al sistema virtual determina procedimientos para llevar a cabo la fragmentación y distribución de la base de datos, con el fin de optimizar sus flujos de información.

Aplicando estos procedimientos, se obtiene una base de datos distribuida con los mejores índices de rendimiento para el entorno en el que se va a desarrollar el sistema de comercio electrónico. A partir de estos procedimientos se define un conjunto de “Directrices de Interoperabilidad de una Base de Datos Distribuida para La Optimización de Sistemas de Comercio Electronico”, cuya eficiencia se comprueba al realizar un conjunto de pruebas sobre un prototipo de sistema.

Adicionalme, hay que tener presente que la optimización es un proceso de monitoreo continuo, con el fin de encontrar problemas de rendimiento, que una vez detectados, con ayuda de las directices planteaas se procede aplicar procesos de reingeniería e ingeniería inversa para realizar los ajustes necesarios para obtener un desempeño consistente y homogéneo del sistema de comercio electrónico.

SUMMARY

INTRODUCCIÓN

Las mismas estrategias de las grandes firmas comerciales en el mercado tradicional están siendo trasladadas al mercado virtual, donde las ventajas y beneficios del posicionamiento y movimiento empresarial se mantienen vigentes. En general podemos decir que comercio electrónico es cualquier forma de transacción comercial en el que las partes interactúan electrónicamente sin intercambio o contacto físico directo. Actualmente, una de las maneras de mantener presencia en Internet es a través del comercio electrónico, donde el *Front Office* encargado de gestionar el *contenido* y la *presentación* representa la imagen de la empresa en Internet, siendo éstas decisivas para la venta y mercadeo, debido a que de la satisfacción que proporcionen harán que el cliente efectúe sus compras.

De acuerdo a la ubicación geográfica de los usuarios, características de procesamiento y almacenamiento de los sitios, especificaciones de la red de comunicación, se debe analizar la conveniencia de fragmentar y distribuir los datos en los diferentes sitios de Internet. Adicionalmente, se debe decidir el nivel de redundancia de cada uno de los fragmentos. Para esto se debe analizar los costos que involucra mantener los fragmentos y sus réplicas, versus los beneficios que ello representa.

La presente investigación proporciona directrices, que permitan mejorar el nivel de interoperabilidad de una base de datos distribuida, minimizando los tiempos de respuesta de la aplicación Web. Esto permite incrementar el rendimiento del sistema e-commerce, ya que la mayor carga se lleva a cabo en la interacción de estos dos componentes (BDD y aplicación Web).

Los costos de la optimización por el lado del hardware son considerables, además, se debe tomar en cuenta que la tendencia en cuanto al contenido es cada vez más exigente, por lo que se deben adoptar medidas alternativas que no necesariamente involucren repotenciaciones de hardware.

PLANTEAMIENTO DEL PROBLEMA

Un sistema de comercio electrónico tiene accesos a nivel mundial, por lo que su rendimiento depende en gran medida de la eficiencia que con que la base de datos responda a los requerimientos de los usuarios. Adicionalmente, se deben analizar las capacidades de la red ya que esto también afecta en un desempeño homogéneo y aceptable del sistema.

En lo que se refiere a la base de datos, se debe analizar la conveniencia de dividirla y/o replicarla para que la información esté lo más cerca posible a sus usuarios y sea lo más fiable en caso de posibles eventualidades en la red.

En consecuencia el problema consiste en mejorar los flujos de información de la base de datos para que los sistemas de comercio electrónico sean una buena herramienta de negocios entre la empresa proveedora y sus consumidores.

OBJETIVOS

OBJETIVO GENERAL

Establecer directrices de interoperabilidad de una base de datos distribuida para la optimización de un sistema de comercio electrónico.

OBJETIVOS ESPECIFICOS

- Determinar la manera que incide el Nivel de Fragmentación en el rendimiento del DDBMS.
- Determinar la manera que afecta la Política de Asignación en la eficiencia del DDBMS.
- Determinar la manera que repercute el Grado de Replicación en la eficiencia del DDBMS.

- Conciliar los índices de Fragmentación, Asignación y Replicación de acuerdo a la naturaleza y entorno del sistema de comercio electrónico.
- Describir los mecanismos utilizados para el flujo de información entre la DDB y la aplicación Web, así como alternativas de configuración hardware y software de un sistema e-commerce.
- Definir directrices ejecutivas, que guíen el proceso de distribución de una base de datos.

JUSTIFICACION

Una buena solución en comercio electrónico que interactúe con un Sistema de Base de Datos Distribuida requiere una considerable investigación de índices de rendimiento, así como profundo conocimiento de los diferentes sistemas de bases de datos, su operación, seguridad, confidencialidad, leyes de legislación, comunicaciones y aplicaciones Web.

El proceso de optimización de todos los componentes permitirá que ellos trabajen mancomunadamente de manera eficiente y en armonía. Hay que tomar en cuenta que un proceso de optimización trae grandes beneficios, los cuales pueden perderse cuando su costo sobrepasa los beneficios alcanzados. Por este motivo esta investigación determina directrices que permita “plantear estrategias adecuadas de distribución y políticas eficientes de acceso a la información para que el sistema de e-commerce sea eficiente”.

Por otro lado, la relevancia científica de este tema es tal, que se ha creado una organización dedicada exclusivamente a su estudio. Así pues, parte de ésta investigación está basados en los estudios realizados con éste grupo del cual se obtuvo valioso apoyo en cuanto a información, asistencia y herramientas de evaluación. **WEB SERVICES INTEROPERABILITY ORGANIZATION (WS-I).**- *“Es una organización, que se caracteriza por promover la interoperabilidad de servicios Web a través de plataformas, sistemas operativos y lenguajes de programación. La organización trabaja a nivel de las industrias y estándares para satisfacer las necesidades de los clientes, proporcionando*

guías, prácticas adecuadas y recursos par el desarrollo de soluciones de servicio Web". Producto de sus investigaciones es "RESOURCES AND GUIDELINES FOR WEB SERVICES INTEROPERABILITY", al cual este trabajo también será enviado como aporte y en reciprocidad a las investigaciones en esta área.

CAPITULO I

SISTEMAS DE BASES DE DATOS DISTRIBUIDAS EN INTERNET

1.1 BASE DE DATOS DISTRIBUIDA (DDB)

Una *Base de Datos Distribuida (DDB.- Distributed Database)* es una colección de datos que pertenecen lógicamente a un mismo sistema, pero que se encuentra físicamente esparcido en varios sitios de la red, conectados entre sí mediante algún tipo de red de comunicaciones, en el cual cada sitio es un sistema de base de datos en sí y si es necesario trabajan juntos con el fin de que un usuario pueda obtener acceso a los datos desde cualquier punto de la red tal como si todos los datos estuvieran almacenados en el sitio propio del usuario.

Las *Bases de Datos Distribuidas (DDB)* realizan dos tipos de transacciones bien diferenciados:

1. *Transacciones Locales:* Cuando se accede a los datos del único sitio donde se inició la transacción.
2. *Transacciones Globales:* Cuando se accede a datos de sitios distintos al emplazamiento donde se inició la transacción.

Partiendo de la definición anterior (DDB), se puede definir a un *Sistema Manejador de Base de Datos Distribuida (DDBMS.- Distributed Database Management System)* como un componente software que permite el manejo de bases de datos distribuidas (DDB), y hace esta distribución transparente a los usuarios. Un ejemplo de este tipo de aplicaciones es el comercio electrónico que utiliza un DDBMS para poder acceder a los datos.

1.2 ARQUITECTURA DE UNA DDB EN INTERNET

Existen muchas alternativas posibles en la implementación de DBMS distribuidos, el método seleccionado define a la vez la arquitectura de distribución de una base de datos distribuida:

- *Arquitectura Cliente / Servidor.*- Cuando múltiples clientes acceden a un solo servidor de bases de datos.
- *Arquitectura Múltiples Clientes / Múltiples Servidores.*- Son más flexibles, ya que la base de datos es distribuida en varios servidores. Cada cliente tiene un servidor directo al cual hace sus peticiones. La comunicación entre los servidores y el cliente es transparente.
- *Arquitectura de Igual a Igual.*- Un DBMS verdadero no distingue entre cliente y servidor. Idealmente cada máquina puede desempeñar la funcionalidad de cliente y de servidor. La base de datos es fragmentada y distribuida físicamente en diferentes lugares.

1.3 METODOLOGÍAS DE DISEÑO

Son un conjunto de procedimientos a seguir para obtener los esquemas físicos de la base de datos. Para abordar el diseño de una base de datos distribuida se tiene dos enfoques o estrategias:

1. *El enfoque Top-Down (o estrategia descendente).*- Es el más utilizado en el diseño de las bases de datos distribuidas desarrolladas desde los inicios de la planeación. Esta investigación utiliza este enfoque por ser más instructivo para nuestro análisis y demostración.
2. *El enfoque Bottom-Up (o estrategia ascendente).*- Es el más utilizado en el desarrollo de los sistemas de base de datos múltiples como una agregación a los sistemas de bases ya existentes, se parte de los esquemas conceptuales locales para llegar al esquema conceptual global.

1.3.1 METODO TOP-DOWN O ENFOQUE DESCENDENTE

En este enfoque el diseñador debe entender los requerimientos del usuario de una aplicación de base de datos, para transformarlas en cada una de las fases en especificaciones formales, las que progresivamente mejoran las especificaciones de alto nivel.

Así, para el diseño de una base de datos centralizada se sigue cuatro fases:

1. Análisis de Requerimientos
2. Diseño Conceptual
3. Diseño Lógico
4. Diseño Físico

Para las bases de datos distribuidas se añade una fase más:

5. Diseño de Distribución.

A continuación se hace una breve descripción de las entradas, procesos y salidas que aportan cada una de las fases de diseño centralizado. Posteriormente se analizará la fase del Diseño Distribuido, la cual es el objetivo de esta investigación.

1. Análisis de Requerimientos.- Define el entorno del sistema para obtener datos como las necesidades de procesamiento de todos los posibles usuarios de la base de datos, los requisitos del sistema, los objetivos que se debe cumplir respecto a niveles de rendimiento, seguridad, disponibilidad y flexibilidad, sin olvidar el aspecto económico. En definitiva se determina “que se va hacer”.

2. Diseño conceptual.- Se encarga de examinar la empresa para determinar los tipos de entidades y establecer la relación entre ellas. Produce una especificación conceptual de un esquema global de base de datos integrada y de las aplicaciones que se llevan a cabo en ella. Se ignora cualquier detalle que tiene que ver con la implementación física, y en particular, la distribución de los datos. El resultado es

un esquema de base de datos global que incorpora en un nivel abstracto todos de los elementos (“conceptos”) y los patrones de su uso. Se realizan estimaciones que giran en torno a la frecuencia de acceso por parte de las aplicaciones, a las distintas relaciones que se hace uso, puede afinarse aún más anotando los atributos de la relación a la que accede. Ejm.

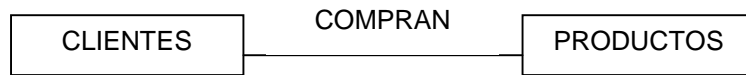


Fig. 1.- Esquema de Diseño Conceptual

3. Diseño lógico.- Transforma el esquema conceptual integrado en un esquema de base de datos de un tipo dado de DBMS, que puede ser relacional, red o jerárquico. La elección del tipo de DBMS depende de los requerimientos de los modelos conceptuales así como de consideraciones pragmáticas.

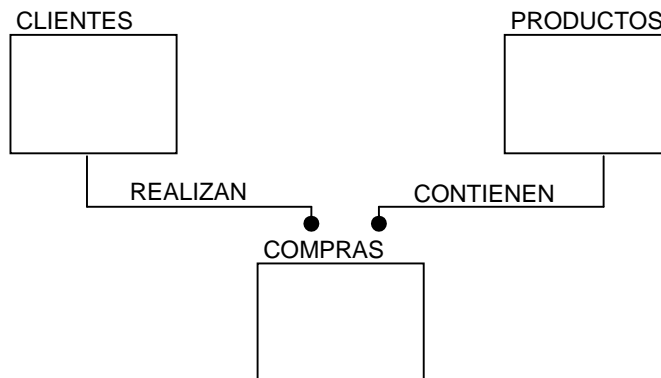


Fig.2.- Esquema de Diseño Lógico

4. Diseño físico.- Se lleva acabo de acuerdo con las capacidades y características del DBMS particular escogido y produce la definición de las estructuras de acceso físico.



Fig. 3.- Esquema de Diseño Físico

5. Diseño de distribución.- A esta fase llega como input un esquema global independiente del sistema y da como resultado los sub-esquemas para cada sitio de la base de datos distribuida. El objetivo de esta etapa consiste en diseñar los esquemas conceptuales locales que se distribuirán a lo largo de todos los puntos del sistema distribuido. Sería posible tratar cada entidad como una unidad de distribución.

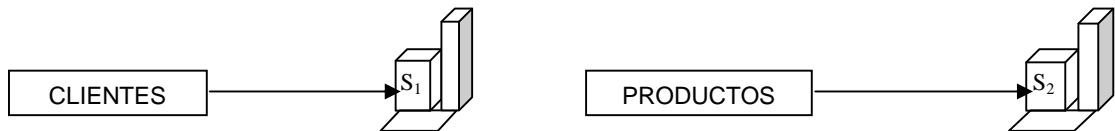


Fig. 4.- Diseño de distribución - asignación

En el caso del modelo relacional, cada entidad se corresponde con una relación. Resulta bastante frecuente dividir cada relación en sub-relaciones menores denominadas fragmentos que luego se ubican en uno u otro sitio.

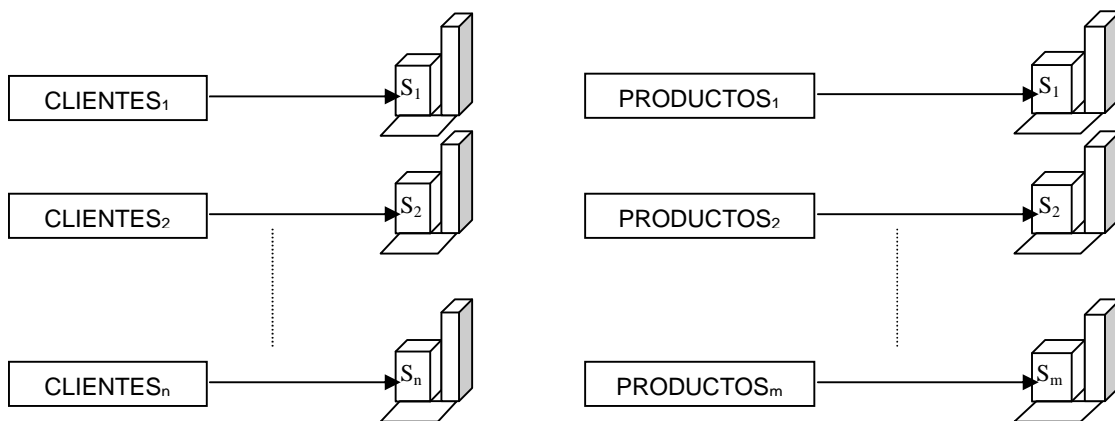


Fig. 5.- Diseño físico - fragmentación

De aquí, que el proceso de diseño de distribución consta de dos actividades fundamentales: la fragmentación y la asignación.

1.3.1.1 DISEÑO DE DISTRIBUCIÓN

Esta es una fase de diseño específica de las bases distribuidas, indica el esquema global y varios sub-esquemas, cada una representa un subgrupo de información que esta asociado con cada sitio. En esta investigación se utiliza el enfoque en dos fases, esto es, primeramente el fraccionamiento y luego se asigna los fragmentos generados a los sitios.

La *fragmentación* es el proceso de subdividir un objeto global, en nuestro caso una entidad o relación en varias partes, llamadas fragmentos. Por su parte la *asignación* es el proceso de ubicación de cada fragmento en uno o más sitios.

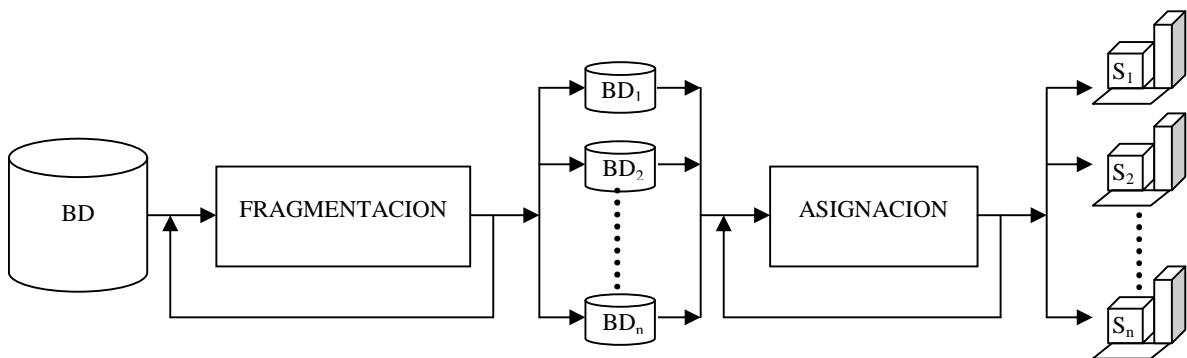


Fig. 6.- Fragmentación - asignación

Esto sugiere que el diseño de distribución debería realizarse o al menos considerarse al principio de la fase de diseño lógico. En ese momento, los datos y las operaciones precisamente se describen y los primeros problemas de la implementación se consideran.

1.3.1.2 MANEJO DE LA FRAGMENTACION

El objetivo de la fragmentación es producir fragmentos con el máximo potencial de localidad con respecto a operaciones. La fragmentación permite poner los datos cerca de los usuarios que los necesitan, de esta forma reduciendo potencialmente el costo de la transmisión.

Una base de datos totalmente fragmentada es aquella donde no existe réplica del fragmento en ninguno de los sitios de la red, los fragmentos se alojan en sitios donde únicamente existe una copia de cada uno de ellos a lo largo de toda la red. En caso de réplica, podemos considerar una base de datos totalmente replicada, donde existe una copia de todo el banco de datos en cada sitio, o considerar una base de datos parcialmente replicada, donde existan algunas copias de los fragmentos ubicados en diferentes sitios.

1.3.1.3 TIPOS DE FRAGMENTACIÓN

1.3.1.3.1 Fragmentación Horizontal.- Se realiza sobre las tuplas de una relación, el resultado consiste en un subconjunto de instancias (tuplas) de un objeto global (entidad). Cada fragmento es un subconjunto de las tuplas de la relación y está asociado con un predicado de cualificación (minitérmino) que indica las propiedades distinguidoras de las instancias o tuplas de ese fragmento.

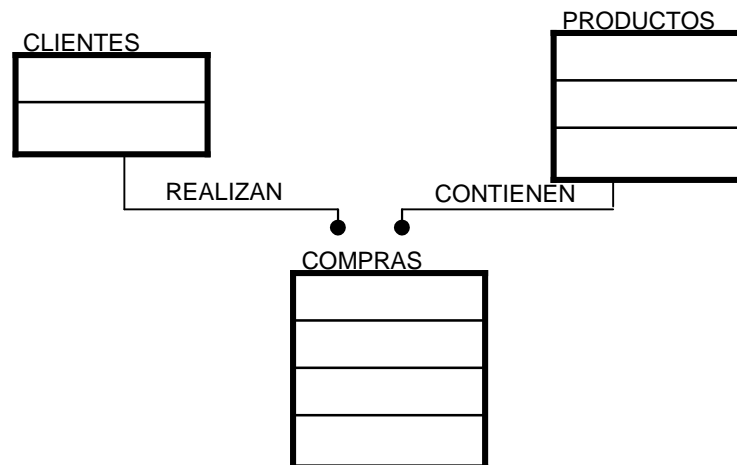


Fig. 7.- Fragmentación horizontal

Existen dos variantes de la fragmentación horizontal:

- a. *La fragmentación horizontal primaria.-* Se desarrolla empleando los predicados definidos en la misma relación (propietaria).

- b. *La fragmentación horizontal derivada.*- Consiste en dividir una relación partiendo de los predicados definidos sobre otra relación (miembro).

1.3.1.3.2 Fragmentación Vertical.- Se derivan de la proyección de los objetos globales (entidades) en subgrupos de sus atributos. Para que la proyección sea más efectiva, se requiere que cada fragmento incluya el atributo clave del objeto global. La fragmentación vertical agrupa atributos frecuentemente usados, una fragmentación vertical ideal existe cuando cada aplicación usa un subset de atributos. Sin embargo, algunas aplicaciones serán demoradas cuando necesiten acceder a fragmentos de arranque que se encuentran en otra ubicación.

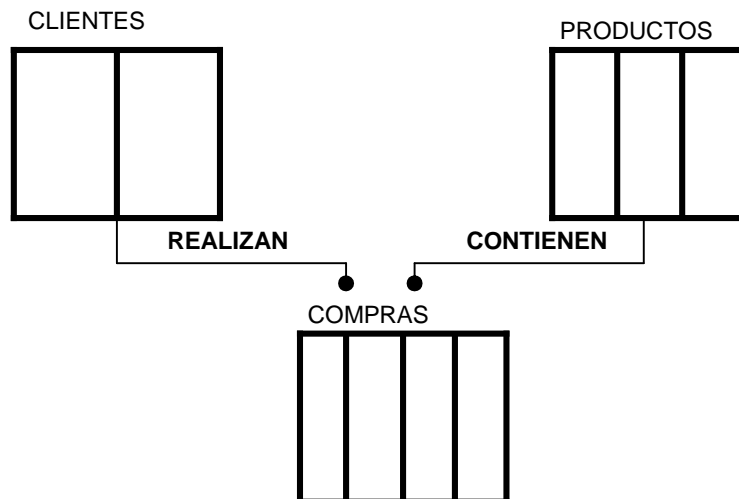


Fig. 8.- Esquema fragmentación vertical

1.3.1.3.3 Fragmentación Mixta o Híbrida.- En algunos casos la fragmentación vertical u horizontal del esquema de la base de datos no será suficiente para satisfacer los requisitos de las aplicaciones. Podemos combinar ambas, obteniendo la denominada fragmentación mixta, que puede llevarse a cabo de tres formas diferentes:

1. Desarrollando primero la fragmentación vertical y, posteriormente, aplicando la partición horizontal sobre los fragmentos verticales. Esta es denominada partición VH.

2. Aplicando primero una división horizontal para luego, sobre los fragmentos generados, aplicar una fragmentación vertical. Esta es denominada fragmentación HV.
3. De forma directa considerando la semántica de las transacciones.

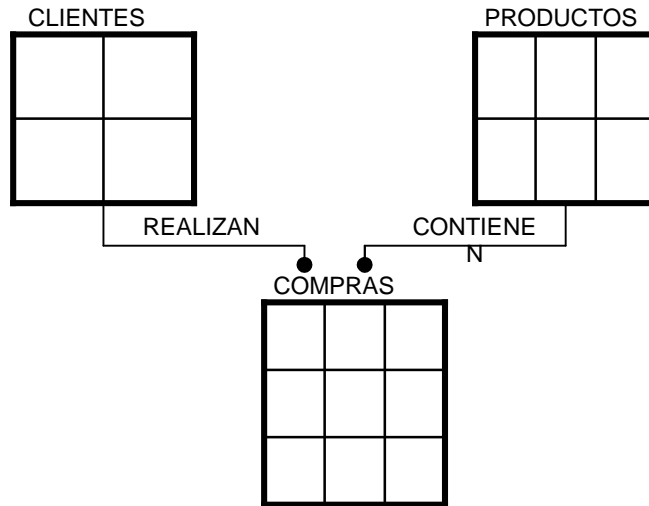


Fig. 9.- Esquema fragmentación híbrida

Tanto a la fragmentación HV como la fragmentación VH, se aplican los mismos criterios y reglas que a la fragmentación horizontal y vertical. Una característica común a ambas es la generación de árboles que representan la estructura de fragmentación.

1.3.1.4 ASIGNACIÓN O COLOCACIÓN

Luego del proceso de fragmentación se debe asignar los fragmentos en uno o varios sitios de la red. Para decidir la ubicación, se debe considerar factores como frecuencia de acceso, tamaño de los fragmentos, capacidad de almacenamiento y procesamiento de cada sitio. Una buena estrategia de asignación permite mayor disponibilidad, incremento del paralelismo, menor tiempo de recuperación.

La asignación de fragmentos puede ser redundante o no redundante. En una asignación redundante cada fragmento es proyectado a uno o más sitios. En una asignación no redundante un fragmento es proyectado exactamente a un sitio.

1.3.1.4.1 ASIGNACIÓN REDUNDANTE

En una asignación redundante el diseñador debe decidir el grado de replica de cada fragmento. Las ventajas de la réplica contraponen los beneficios de la recuperación con los de las actualizaciones. Por un lado la aplicación accede a la copia más cercana, disminuyendo el tiempo de espera, además que aumenta su disponibilidad ya que en caso de que un fragmento se aíse por algún motivo, se puede recurrir a su réplica en otro sitio.

La desventaja es el incremento de los costos de almacenamiento, ya que las copias que son replicadas requieren de más espacio, además que cuando se realiza actualizaciones, éstas se deben proyectar a cada una de las copias, lo que incrementa el tiempo de dicho proceso de acuerdo al número de réplicas que tenga que actualizarse, lo que conlleva a que el sistema este temporalmente inconsistente.

1.3.1.4.2 ASIGNACION NO REDUNDANTE

La ventaja de este tipo de réplica consiste en que alcanza la máxima consistencia de la información, así como se optimiza al máximo el espacio de almacenamiento debido a que solo una copia de la información esta presente en toda la red. Sin embargo, se debe analizar el costo en cuanto al nivel de aislamiento o bloqueo a los fragmentos cuando requieran operaciones de recuperación y actualización.

La principal desventaja es el riesgo a la vulnerabilidad que se corre en caso de que un determinado sitio salga accidentalmente de la red.

1.3.2 METODO BOTOM-UP O ENFOQUE ASCENDENTE

Este enfoque asume que la especificación de las bases de datos ya existe, ya sea por que hay bases de datos que tiene que ser interconectadas a un sistema de bases múltiples, o porque la especificación

conceptual de las bases ha sido hecha para cada sitio independientemente. En cualquiera de los dos casos, las especificaciones de los sitios se deben integrar para generar una especificación global.

En este escenario, se deben analizar toda la documentación que forma parte del sistema de comercio electrónico. Para establecer la heterogeneidad existente entre:

- los diferentes sistemas de bases de datos que lo conforman,
- la variedad de sistemas operativos que interactúan,
- la multiplicidad de aplicación que se integran,
- los distintos equipos de computación que se utilizan, y
- la diversidad de medios de comunicación en los que trabajan

Ya que el objetivo de esta investigación es analizar la interoperabilidad de una base de datos distribuida en un sistema de comercio electrónico, y más no analizar los modelos de desarrollo, esta investigación utiliza el enfoque Top-Down.

1.4 PROGRAMAS CGI COMO INTERFAZ PARA ACCEDER A BASES DE DATOS DISTRIBUIDAS

Para la integración de la base de datos con la aplicación Web es necesario contar con una interfaz que:

- realice las conexiones
- extraiga la información de la base de datos
- formatee la información para ser visualizada en un navegador Web, y
- permita lograr sesiones interactivas entre la base de datos y la aplicación, permitiendo que el usuario seleccione la información que requiere.

El medio que permite crear esta interfaz es un componente denominado *CGI (Common Gateway Interface)* que es externo al navegador Web.

Los programas CGI son Gateway de doble sentido. Los datos pueden transferirse al programa CGI para su procesamiento, al igual los programas CGI pueden devolver información al servidor Web. Con esto la información introducida por el usuario puede afectar el comportamiento del programa CGI, y lo devuelto por el programa son resultado directo de lo que introduce el usuario. En algunos casos los programas CGI se ejecutan al cargar la página y los resultados se despliegan como parte de la misma.

Un *Gateway* es una conexión con el sistema operativo externo, la acción de llamar a un programa CGI desde un navegador Web es muy fácil para el usuario, lo cual es uno de sus principales atractivos. El tener acceso a una base de datos a través de un programa CGI tiene una metodología propia, comúnmente el usuario hace clic sobre un botón predeterminado o sobre un vínculo, en este momento el navegador envía una solicitud CGI al servidor Web para ejecutar el programa, el servidor Web revisa la configuración y los archivos de acceso para asegurarse que se cuenta con el permiso de ejecución del programa CGI y se asegura que este exista, cualquier resultado producido por el programa CGI se devuelve al navegador Web que despliega el resultado.

Los Gateway reciben datos transferidos desde un navegador Web a través de un servidor de HTTP y los convierten a un formato que la base de datos pueda entender. La información convertida se transfiere a la *Interfaz de la Base de Datos* y esta la ejecuta. Los resultados se devuelven al Gateway, el cual los convierte a un formato que el navegador los pueda desplegar.

La *Interfaz de la Base de Datos* es un conjunto de controladores *ODBC* que crea el enlace o canal con el *DBMS*. Es un software especial que suministra el proveedor de la base de datos, en varios niveles de complejidad dependiendo de las necesidades. Entonces, el Gateway se comunica con la Interfaz, la cual se pone en contacto con la base de datos.

En definitiva, con CGI se puede:

- Crear páginas Web con acceso a información contenida en bases de datos.
- Insertar, actualizar o eliminar información de la base de datos según la entrada del usuario en una página Web.
- Ejecutar otros comandos *SQL (Structured Query Language)*.

Dependiendo de las necesidades de la aplicación, el diseño de la base de datos puede o no tener capacidades de redes (distribución), esto lo determina el software del proveedor. El método más sencillo se alcanza con una base de datos sin capacidad de red, pues el servidor de HTTP, los programas CGI y el servidor de la base de datos están ubicados en la misma máquina (Ver Fig. 16).

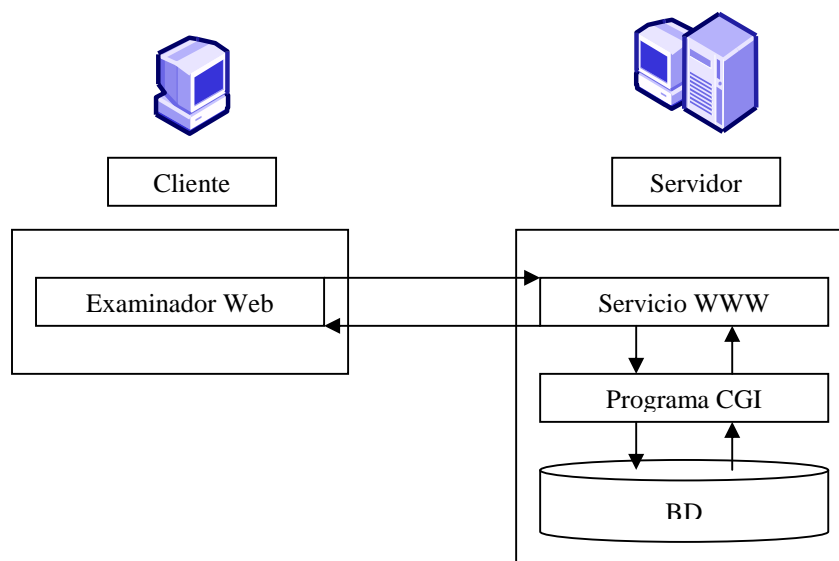


Fig. 10.- Esquema servidor HTTP, BD, CGI

De esta forma los programas CGI tienen acceso a cualquier programa de interfaz que deseen utilizar, las consultas y los resultados devueltos no van y vienen a los programas CGI a través de la red, disminuyendo el tiempo de respuesta. Sin embargo, para esta investigación, se supone que los navegadores están accediendo a la información de manera remota.

1.5 PROCESO DE EXPLORACIÓN EN UNA DDB CON CGI

En el proceso de exploración de una base de datos los componentes mencionados anteriormente intervienen para transferir la consulta del usuario a la base de datos y devolver los resultados al navegador. Podemos describir este proceso en los siguientes pasos:

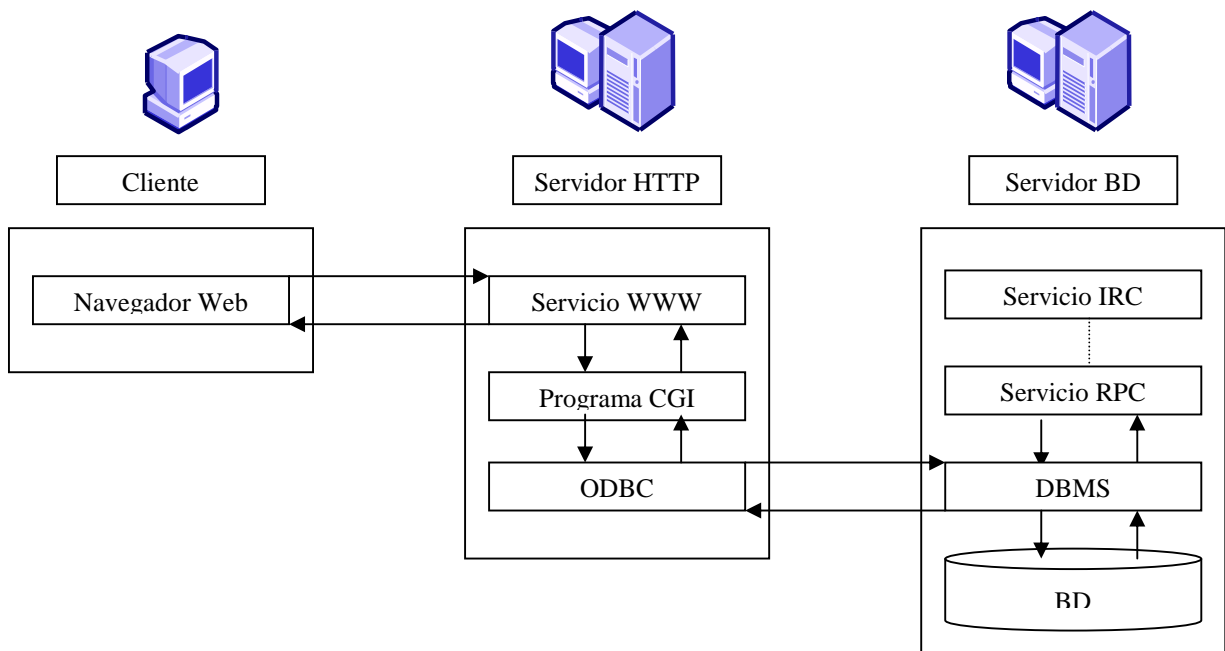


Fig. 11.- Esquema exploración de BD con un CGI

1. En el navegador, el usuario llama a un programa CGI, haciendo clic en un Hipervínculo u oprimiendo un botón del formulario.
2. El navegador reúne toda la información escrita por el usuario para enviarla al programa CGI.
3. El navegador contacta al servidor de HTTP en la máquina donde reside el programa CGI, pidiéndole que localice a este último y le transfiera la información.
4. El servidor de HTTP corrobora si el usuario solicitante tienen autorización de acceso al programa CGI.
5. Si el usuario tiene autorización de acceso, el servidor de HTTP localiza el programa CGI y transfiere a este la información del navegador Web.

6. Se ejecuta el programa CGI.
7. El proceso Gateway convierte la información recibida a un formato que la base de datos sea capaz de entender.
8. El Gateway usa el módulo de la base de datos para transferir la consulta a la interfaz de la base de datos.
9. La interfaz de la base de datos analiza la sintaxis de la consulta para asegurar que sea precisa.
10. Si la interfaz encuentra un error de sintaxis en la consulta, se envía un mensaje de error al Gateway.
11. El mensaje de error se envía al servidor de HTTP, el cual lo transfiere al navegador Web para que este lo despliegue al usuario.
12. Si no hay error, la interfaz envía la consulta a la base de datos.
13. La base de datos atiende la consulta y devuelve los resultados al Gateway a través de la Interfaz.
14. El Gateway formatea los resultados y los envía al servidor, por medio del CGI, para su envío al navegador Web.
15. El navegador Web despliega los resultados

Hacer inserciones, actualizaciones o eliminaciones requiere de un procedimiento mas complejo, ya que el código de inserción debe verificar si existen los datos en la base de datos, con el fin de evitar duplicidad en la información, el código de actualización debe asegurarse que la información exista, antes de modificarla, en caso contrario debe originarse un error. Las acciones de eliminación y modificación deben asignarse a un grupo reducido y confiable de usuarios, para evitar la eliminación de datos necesarios. Para estos casos, es necesario algún tipo de control de acceso.

1.6 MANEJO DISTRIBUIDO DE TRANSACCIONES

El manejo de transacciones tiene dos aspectos principales, el control de recuperación y el control de concurrencia, cada uno de los cuales requiere un tratamiento más amplio en el ambiente distribuido. Para explicar ese tratamiento es preciso introducir primero un término nuevo, "agente". En un sistema distribuido, una sola transacción puede implicar la ejecución de código (actualizaciones) en varios sitios. Por tanto, cada transacción está compuesta de varios agentes, donde un agente es el proceso ejecutado en nombre de una transacción dada en determinado sitio.

La cuestión específica al control de recuperación para asegurar que una transacción dada sea atómica (todo o nada) en el ambiente distribuido, debe asegurar que todos los agentes correspondientes a una transacción se comprometan al unísono o bien que retrocedan al unísono. Este efecto puede lograrse mediante el protocolo de compromiso en dos fases.

En cuanto al control de concurrencia, esta función en un ambiente distribuido está basada en el bloqueo, al igual que en los sistemas no distribuidos. El sistema necesita saber cuándo dos agentes son parte de la misma transacción, ya que no puede permitirse un bloqueo mutuo entre dos agentes que sean parte de la misma transacción. Esto es manejado a través de protocolos.

1.7 PROTOCOLOS DE CONTROL DE CONCURRENCIA

En sistemas distribuidos las operaciones de una transacción dada deben ejecutar múltiples sitios cuando ellas accedan a los datos. Un efecto secundario de los algoritmos de control de concurrencia basados en el bloqueo es que ellos causan los llamados *deadlock*. Para minimizar que la base de datos pueda quedar en algún momento en este estado se utilizan protocolos de control de concurrencia como:

Protocolos de seguridad.- Los protocolos de seguridad distribuida imponen la atomicidad, que es una propiedad de todo o nada de transacciones implementando protocolos atómicos que aceptan los cambios tales como el commit de dos fases en el que todas las máquinas involucradas en la ejecución de una transacción distribuida aceptan hacer la transacción antes de que sus efectos sean hechos permanentemente, por lo que todas las máquinas terminan la transacción de la misma manera.

Protocolos de recuperación.- Los protocolos de recuperación distribuida tratan con el problema de la recuperación de una base de datos de una máquina que ha fallado llevándola a un estado consistente cuando el equipo se recupera de la falla.

Protocolos de replica.- El problema en este tipo de bases de datos es el de mantener algún tipo de consistencia entre las instancias a las copias físicas cuando los usuarios transparentemente actualizan lógicamente los datos. Si hablamos de consistencia hablamos de que cualquier copia es equivalente a las otras, de manera que los valores de todas las copias físicas de los datos lógicos deben ser idénticos cuando la transacción de actualización ha terminado. Un protocolo de control de replica es conocido como lee uno y escribe en todas (ROWA).

CAPITULO II

SISTEMAS DE COMERCIO ELECTRONICO

Es importante conocer el origen desarrollo y evolución de los sistemas de comercio electrónico para comprender el gran potencial que este tipo de negocios representará para el futuro de las economías, en tal virtud y para que este estudio sea completo en el Anexo I, se hace un estudio minucioso de estos aspectos.

Si bien es cierto que todos los procedimientos son importantes, el manejo de la Base de Datos Distribuida conjuntamente con el Pago es probablemente la parte más compleja desde el punto administrativo y operativo del comercio virtual. La base de datos distribuida porque del proceso transaccional depende su consistencia. El pago porque la parte económica es vital en toda transacción.

En la etapa actual B2B de comercio electrónico se deben canalizar considerables flujos de información en las transacciones entre la empresa (proveedor) y sus interlocutores de negocio (clientes, financieras, etc.), por lo que es necesario conocer todos los aspectos que pueden influenciar su rendimiento.

2.1 EL MODELO FUNCIONAL CLIENTE - SERVIDOR DE INTERNET

Cuando se utiliza Internet, se establece un proceso, en el cual interactúan dos partes. Por un lado, el usuario ejecuta una aplicación en la computadora local, que es el denominado programa *cliente*, este programa se pone en contacto con el host remoto para solicitar la información deseada. El host remoto

a su vez, responderá al pedido realizado por el programa cliente, a través de otro programa de aplicación, denominado el programa *servidor*.

Los términos *cliente* y *servidor* se usan tanto para referirse a los programas que cumplen estas funciones, como a las computadoras donde son ejecutados dichos programas.

2.1.1 PROGRAMA CLIENTE (NAVEGADOR WEB)

Cuando se escribe una dirección URL en el navegador, es proyectada a una dirección y puerto IP correspondiente a un servidor Web. Después de haber establecido la conexión, el cliente y el servidor se comunican con el Protocolo de Transferencia de Hipertexto (HTTP).

El programa *cliente* cumple dos funciones distintas:

1. Gestiona la comunicación con el servidor, solicita un servicio y recibe los datos enviados por el *servidor*; y
2. Maneja la interfaz con el usuario, presenta los datos en el formato adecuado y brinda las herramientas y comandos para que el usuario pueda utilizar las prestaciones del servidor de forma sencilla.

2.1.2 PROGRAMA SERVIDOR (SERVIDOR WEB)

Un servidor Web es un programa que se ejecuta en una computadora conectada a Internet, donde su trabajo es escuchar en un puerto TCP/IP predefinido las solicitudes de clientes y luego responder a navegadores Web con contenido basado en esas solicitudes.

El *servidor* cumple las siguientes funciones:

1. Acepta peticiones recibidas a través de la red
2. Atiende el servicio y
3. Regresa el resultado al cliente

Con este modelo un mismo *servidor* puede atender a varios *clientes* al mismo tiempo, sin embargo el servidor procesa una petición a la vez. El sistema operativo hace una cola de peticiones para el servidor mientras este está ocupado.

Los servidores tienen dos partes importantes:

1. *Un programa maestro.*- Es responsable de aceptar nuevas peticiones, y
2. *Un conjunto de esclavos.*- Son responsables de manejar las peticiones individuales

Como el maestro ejecuta un esclavo para cada nueva petición el procesamiento procede de manera concurrente, de este modo las peticiones que requieren de poco tiempo para completarse pueden terminar antes que las peticiones que llevan más tiempo en ejecución, independientemente del orden en que hayan comenzado.

2.2 INFRAESTRUCTURA PARA EL E-COMMERCE

Como medio para que los procesos transaccionales se lleven a cabo se utiliza el Internet, el mismo que proporciona la infraestructura de comunicación para el comercio electrónico, mediante el cual, el *Cliente* y la *Empresa* pueden realizar transacciones comerciales B2B (business to business).

En cuanto al software la estandarización tecnológica de Internet está basado en un conjunto de protocolos llamados *TCP/IP* (*Transmission Control Protocol/Internet Protocol*). Están diseñados para satisfacer la necesidad de un sistema de comunicación interoperable para el enlace de redes. El diseño de los protocolos TCP/IP, realiza las conexiones de forma transparente al usuario sin importar la ruta que la conexión establezca, de esta forma se oculta al usuario la arquitectura de la red global

Un sistema de comercio electrónico utiliza:

- información (DB)
- administrador de información (DBMS)
- canalizadores de información (CGI)
- programas de acceso a información (Aplicación)
- medio de comunicación (Internet)

2.3 CONSIDERACIONES EN EL DESARROLLO DE APLICACIONES E-COMMERCE

Para la construcción de aplicaciones de comercio electrónico se puede utilizar cualquiera de las metodologías de desarrollo de software, sin preferencia en particular por alguna de ellas, mas bien esto estará supeditado a la experiencia y conocimiento del ingeniero en software. En tal virtud, en este capítulo no se trata una metodología en particular, por salir del objetivo de esta investigación, pero sí se resalta los aspectos inherentes a sistemas de esta naturaleza que se deben considerar con cualquier metodología que se use.

2.3.1 ANÁLISIS PRELIMINAR

Uno de los puntos neurálgicos en el desarrollo de cualquier tipo de sistemas, es precisamente sus fases iniciales de levantamiento y documentación de información. Los *SRS (Software Requirements Specification)* precisamente buscan minimizar los problemas de comunicación e interpretación que puedan darse con el equipo de desarrollo.

Para sistemas de esta naturaleza se puede utilizar como guía un documento denominado *Reglas Comerciales*, que se describe mas adelante, el cual incluye cuestiones relativas al e-commerce que se deben tener en cuenta a la hora de elaborar un plan pormenorizado para la creación de un sitio.

Es recomendable establecer prioridades, ya que es probable que no se pueda implementar todo lo que se desee en la primera versión del sitio, debido a restricciones económicas o técnicas, pero, se puede implementar aquellas funciones que permitan mantener al sistema mientras las demás se van integrando progresivamente.

2.3.2 REGLAS COMERCIALES

Permiten resaltar las peculiaridades de sistemas de esta naturaleza. Utiliza un cuestionario cuyas respuestas proporcionan útil información a tener en cuenta en el momento del desarrollo. Los aspectos a considerar se detallan en el Anexo II. Con la información provista por este documento, se puede analizar la complejidad del sitio de e-commerce ya que ayuda a describir las técnicas, procesos e índices deseados de funcionalidad del sitio Web.

2.4 ARQUITECTURA DEL SISTEMA DE COMERCIO ELECTRÓNICO

La arquitectura de un sistema de comercio electrónico, define los componentes básicos e importantes, así como la descripción de las relaciones que existe entre ellos. La arquitectura, es la manera en la cual se agrupan las funciones en unidades, se puede determinar sí:

- Todos los componentes están integrados en un solo sistema
- Los componentes están distribuidos en diferentes sistemas
- Cuáles son las interfaces entre las unidades funcionales.

Los dos aspectos a considerar en la arquitectura de cualquier sistema son:

1. Los roles
2. La descomposición funcional

Un aspecto adicional en la arquitectura de un sistema de comercio electrónico es:

3. Proceso transaccional del contenido.- Es la manera en que el contenido se encuentra relacionado con el proceso transaccional.

La arquitectura de cualquier sistema computacional complejo y más aún de un sistema de comercio electrónico que depende de varios componentes externos (Internet) es crucial para un buen funcionamiento a largo plazo. No es recomendable desarrollar pequeñas aplicaciones para resolver problemas específicos temporalmente, ya que esto conlleva a un rápido deterioro y obsolescencia del sistema en general, impidiendo planificaciones futuras sobre dicha estructura. Por tal motivo es imprescindible elaborar un diseño que se ajuste a los requerimientos actuales, pero con miras a posibles cambios o adiciones de requerimientos en el futuro.

2.4.1 ROLES Y FUNCIONES

Para poder definirlos se debe contestar estas preguntas:

- *Roles:* Quién lo va a usar? Esto ayudará a enfocar la atención para que cada tipo de usuario.
- *Funciones:* Qué va a hacer con él? Esto permitirá descomponer el sistema en unidades funcionales, así como las interfaces entre ellas.

2.4.1.1 ROLES

Se puede clasificar de acuerdo a las funciones que desempeñan las personas:

- Los compradores
- Los diseñadores de catálogos
- Los representantes de servicio al cliente, y
- Los operadores del sistema

Es necesario que desde el inicio de su diseño se considere los roles de manera independiente del tamaño o estructura organizacional de la empresa, aunque en la práctica estos sean llevados a cabo por una sola persona. Esto permite satisfacer los requerimientos actuales y futuros de la empresa.

2.4.1.2 ROLES DEL COMPRADOR O CLIENTE

En toda transacción comercial siempre hay un vendedor y un comprador. A este último se lo llama también: cliente, consumidor, agente de compras, etc. En lo que a Internet se refiere se denomina cliente o navegador (browser) refiriéndose más al software que a la persona misma.

Los roles que se pueden identificar de acuerdo a la función que realizan por parte del comprador son:

- *Specifier.*- Es la persona que selecciona lo que va a ser comprado
- *Approver.*- Es una persona que aprueba una compra recomendada por el Specifier
- *Buyer.*- Es la persona que negocia los términos y condiciones de una compra y hace los arreglos para el pago
- *Recipient.*- Es la persona que recibe los bienes y servicios entregados

Los compradores también pueden clasificarse de acuerdo a la relación con el vendedor:

- *Walk-in customer (Comprador anónimo o cliente casual).*- Es un comprador nuevo que no tiene relación previa con el vendedor y que no realizará más de una compra
- *Member client (Cliente de membresía).*- Es un comprador que ya tiene un historial de compras con el vendedor. Para esta membresía el comprador debe subscribirse por periodos y como contraparte el vendedor le da beneficios respecto a los demás.
- *Member administrator (Administrador de suscriptores).*- Es un cliente de membresía que representa a un grupo de cuentas individuales, el cual tiene capacidad de actualizar a sus miembros. El director de compras de una empresa que tiene membresías para cada uno de sus compradores.

En la práctica una sola persona puede realizar todos los roles:

- Seleccionar el artículo
- Pagar el costo del artículo
- Llevarse el artículo a su casa

En una empresa puede haber un responsable para cada uno de estos roles:

- *El conocedor del producto.*- Determina la mercadería que hay que comprar
- *El agente de compras.*- Negocia los términos de pago
- *El personal de manufactura.*- Recibe la mercadería para fabricar el producto final

2.4.1.3 ROLES DEL VENDEDOR O PROVEEDOR

La otra parte de la transacción comercial esta representada por el vendedor. Existen dos grupos principales de roles:

El equipo de creación de contenido y comercialización:

Bussines manager (administrador empresarial).- Se encarga de dar la presencia empresarial de la empresa en Internet. Define los productos, sus precios, las relaciones con otras empresas.

Internet commerce architech (arquitecto del comercio electrónico).- Tiene el perfil de un analista de sistemas, capaz de poder interpretar e implementar: nuevos requerimientos comerciales, el procesamiento de la transacción, el cumplimiento del trato y los aspectos técnicos del servicio al cliente.

Content designer (diseñador del contenido).- Es responsable de que el usuario este satisfecho con el diseño en cuanto al despliegue de información

Content author (autor de contenido).- Se encargada de diseñar la información a cerca del producto a ser presentada

Implementor (implementador).- Es el responsable de desarrollar el software necesario para que el sistema de e-commerce interactúe dinámicamente con la base de datos

Database administrator (Administrador de la base de datos).- Se encarga de asegurar la integridad, disponibilidad, corrección y funcionamiento de la base de datos

Sales and marketing (Ventas y mercadeo).- Es responsable de promover comercio electrónico de la empresa

Customer service representative (Servicio al cliente).- Es quien proporciona el servicio de post-venta, información adicional respecto a: los productos, proceso de registro o compra, responde a las quejas, por lo que debe tener especialistas en cada área.

El equipo operativo.- Instala y opera el sistema de comercio electrónico, manteniéndolo disponible y correctamente funcionando.

Operations manager (administrador de operaciones).- Se encarga de administrar todos los servicios del sistema de comercio electrónico.

System supervisor (supervisor del sistema).- Es responsable de administrar el personal del sistema.

System administrator (administrador del sistema).- Es el encargado de la operación técnica del sistema y de las redes.

Security officer (responsable de seguridad).- Se encarga de que las seguridades estén tanto en el diseño como en la implementación del sistema.

Fulfillment agent (agente de cumplimiento de contrato).- Es el responsable de supervisión y operación de los envíos o entregas de los bienes o servicios.

Accountant (contador).- Se encarga de vigilar que los procedimientos contables sean los adecuados para las operaciones comerciales, llevando los respectivos registros.

2.4.2 COMPONENTES FUNCIONALES

Los componentes funcionales son el conjunto de aplicaciones software que utiliza tanto el comprador como el vendedor para poder llevar a cabo una transacción comercial.

2.4.2.1 COMPONENTES DEL COMPRADOR O CLIENTE (NAVEGADOR)

El comprador utiliza el navegador, browser o cliente Web para realizar su rol. Actualmente la arquitectura del sistema depende básicamente de la estructura del Web y particularmente de las capacidades del navegador. Para abordar el diseño es necesario analizar dos aspectos:

1. El navegador con que cuenta el comprador
2. Las capacidades del navegador

Se han desarrollado aplicaciones especializadas llamadas “wallets”, para realizar y llevar el registro del estado del pago, transacciones, órdenes, etc. El principal inconveniente es que en caso de que éste deba estar en el equipo del cliente, él deberá conseguir e instalar dicho wallet.

2.4.2.2 COMPONENTE DEL VENDEDOR O PROVEEDOR (SERVIDOR)

El vendedor debe proporcionar todos los componentes de la cadena de valor comercial, desde el contenido hasta el servicio post venta. En la realidad el vendedor proporciona algunos de ellos, y los demás deben contratarlo con otras empresas.

Entre los componentes del vendedor tenemos:

Sistema de procesado de la transacción.- Mantiene un registro a cerca del estado de todas las transacciones. Ej. Que se ordenó, quién ordenó, cuanto cuesta, estado de pago, estado entrega, etc.

Procesadores de pago.- Se encarga de administrar y controlar el dinero o los instrumentos de pago. Ej. En el pago con tarjeta verificar la disponibilidad de los fondos.

Sistemas de entrega.- Por lo general son otras compañías las que se encargan de la entrega, las cuales por lo general están conectadas mediante un sistema similar al vendedor.

Una conjunción lógica de estas funciones da lo que se denomina “front office” y “back office”:

Front office.- Se refiere a la imagen de la empresa en Internet, el contenido y la presentación son muy importantes para la venta y mercadeo. Se trata de atraer al cliente para que efectúe compras.

Back office.- Se refiere al control de la transacción, desde que se elaboró el pedido hasta que la entrega se efectúa y el pago es recibido.

2.5 PROCESO TRANSACCIONAL DEL CONTENIDO CON CGI

El explorador de un cliente Web puede iniciar una aplicación CGI completando un formulario HTML o haciendo clic en un vínculo de una página HTML del servidor Web. La aplicación CGI acepta la información escrita por el usuario, la procesa, y después devuelve los resultados en una página HTML o envía la información a una base de datos. La siguiente ilustración muestra cómo intercambian información un explorador, un servidor y una aplicación CGI.

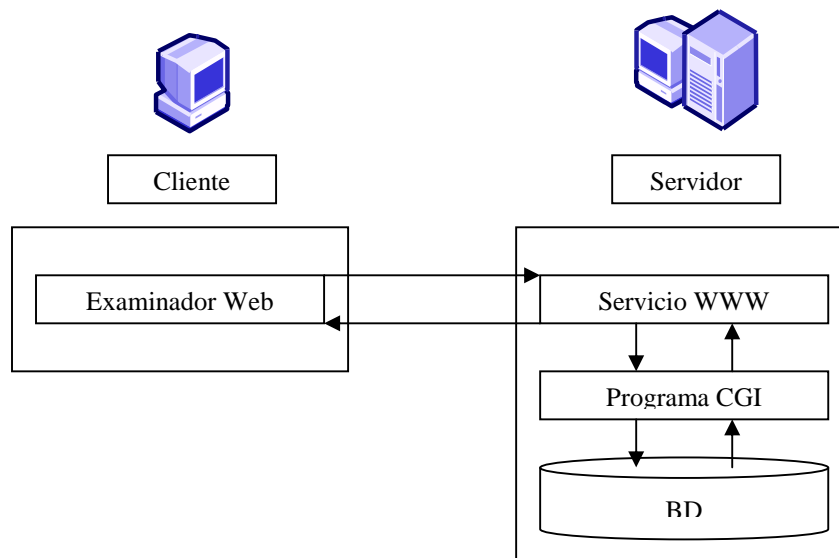


Fig. 12.- CGI en el proceso transaccional

1. *El cliente envía una petición.*- El explorador de un cliente puede realizar una petición CGI a un servidor mediante uno de dos métodos: El cliente inicia un proceso CGI haciendo clic en cualquiera de los elementos siguientes de una página HTML:

- Un vínculo de hipertexto que ejecute el archivo de comandos directamente.

- El botón "Enviar" de un formulario HTML.
 - Un objeto en línea recuperado con el método GET.
 - Un objeto de búsqueda (es decir, uno que utiliza la etiqueta ISINDEX de HTML).
2. *El servidor recibe la petición.*- La dirección URL que el explorador del cliente envía al servidor contiene el nombre del archivo de comandos CGI o la aplicación que va a ejecutarse. El servidor compara la extensión del archivo con la clave de registro ScriptMapping del servidor para decidir que ejecutable debe iniciar.
 3. *El servidor pasa la petición a la aplicación.*- El servidor pasa la información a la aplicación CGI utilizando variables de entorno y, a continuación, inicia la aplicación. Algunas de estas variables están relacionadas con el servidor pero la mayoría vienen del explorador del cliente y tienen relación con el explorador del cliente o con la petición que se está enviando.
 4. *La aplicación CGI devuelve los datos al servidor.*- La aplicación realiza su procesamiento. Si son adecuados, la aplicación escribe los datos en un formato que el cliente pueda recibir en el flujo de salida estándar (STDOUT).
 5. *El servidor devuelve los datos al cliente.*- El servidor toma los datos que recibe de STDOUT y agrega encabezados HTTP estándar y, a continuación, devuelve el mensaje HTTP al cliente. Para mayor detalle ver el Capítulo II.

En lo que al pago respecta hay muchas opciones, como: tarjetas de crédito, débito, órdenes de compra, cuentas corrientes, etc. Debido a que la mayoría de las transacciones de comercio electrónico se realiza a través de tarjetas de crédito, a continuación se describe este proceso.

Las operaciones con tarjeta de crédito son bastante complejas por el número de entidades distintas que intervienen (Ver Fig. 19): la empresa, su propio banco, los clientes, los bancos de los clientes, las empresas emisoras de las tarjetas de crédito y los grandes bancos que "compran" créditos y débitos y que de algún modo determinan todas las transacciones.

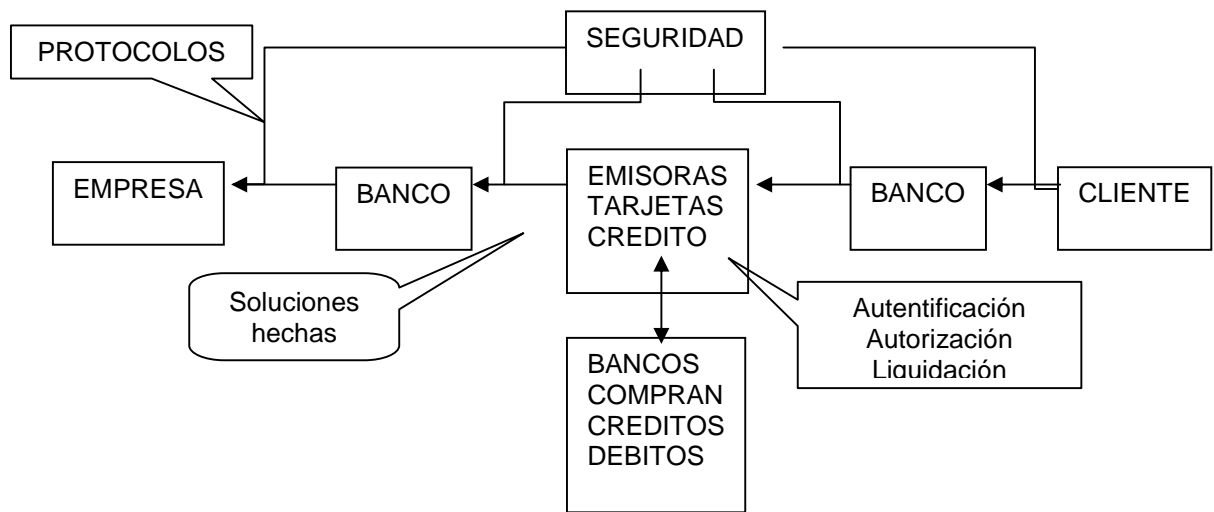


Fig. 13.- Operaciones con tarjeta de crédito.

Podemos resumir las transacciones con tarjetas de crédito en 3 pasos:

1. *Autenticación.*- Verifica que las tarjetas que se presentan tengan un número válido, hayan sido emitidas efectivamente por las empresas correspondientes y no estén denunciadas como robadas.
2. *Autorización.*- Verifica si la tarjeta tiene fondos suficientes para la compra. Si los tiene, se hace una reserva de esos fondos. Pero, aún no se transfiere el dinero.
3. *Liquidación.*- Una vez enviados o entregados los productos al cliente, se pone en conocimiento de este hecho a los bancos involucrados en la operación. Los bancos liberan entonces los fondos reservados previamente y el dinero inicia su recorrido a través de muchos otros bancos e intermediarios hasta llegar a la cuenta de la empresa.

En el mercado hay algunas aplicaciones orientadas a manejar estas situaciones, como CyberCash, Verifone, CyberSource. Para esto, es recomendable utilizar aplicaciones de propósito general ya existentes por tres razones:

1. No se necesitan desarrollar nuevamente
2. Como son distribuidas a gran escala, no es necesario invertir en nuevos canales de distribución
3. Los clientes ya conocen el funcionamiento de estas aplicaciones

2.6 TRANSFERENCIA DE PARAMETROS ENTRE EL WEB Y UN PROGRAMA CGI

Existen diferentes métodos para transferir información de un navegador Web a un programa CGI, esto depende de cómo se llame al programa en el documento HTML. Estos pueden ser:

1. *Por parámetros en la línea de comandos.*- El lenguaje HTML proporciona un método sencillo en la línea de comandos llamado ISINDEX, que debe escribirse dentro de la sección <HEAD> del documento., es la única manera de enviar parámetros de esta forma a un programa CGI.

2. *Por transferencia de variables de entorno.*- Las variables de entorno permanecen desde que el Shell de comandos se activa hasta el momento en que se desconecta de la máquina. Un servidor Web utiliza las variables de entorno y establece los valores de distintas variables cada vez que se llama un programa CGI. El programa CGI puede acceder a esas variables y usarlas dentro del programa. Por ejemplo una búsqueda en yahoo, se traduce en: `http://av.yahoo.com/bin/search?p=marco+yambay`
El URL informa al programa CGI receptor que la variable *p* debe recibir el valor "marco yambay".

3. *Por variables dependientes del protocolo http.*- La interfaz define una serie de variables específicas para servidores de información basados en el protocolo HTTP.

4. *Por entrada estándar.*- Esta es la manera en como los formularios HTML envían datos capturados por medio de listas de selección, casillas de verificación, cuadros de entrada de texto y otras características interactivas disponibles al servidor de HTTP y al programa CGI.

2.7 ELABORACION DE UN MODELO CONFIABLE

En los sistemas distribuidos existe alta dependencia entre los elementos que lo conforman. Las especificaciones de las relaciones entre estos elementos son llamados modelos confiables del sistema.

Todo sistema debe tener al menos un modelo confiable implícito, el mismo que debe estar especificado para analizar la seguridad del sistema a partir de las relaciones entre los componentes.

Para su elaboración se debe analizar factores de riesgo de cada uno de los elementos y relaciones que conforman el modelo. Así pues, para sistemas transaccionales críticos se puede optar por tener sistemas de recuperación y/o switcheo automáticos. Se puede elaborar un cuadro estadístico de factibilidad, impacto, propagación y alcance del riesgo y definir el modelo confiable.

Los principales elementos que se deben considerar son:

- Servidor Web
- Servidor de Aplicaciones
- Servidor de Bases de Datos
- Servidores de Respaldo
- Comunicaciones

2.8 SEGURIDAD EN LA INTERCONEXION DE LA WEB CON LA BD

La seguridad de las transacciones esta orientada a minimizar la posibilidad de que los hackers ataquen, evitando que accesos fraudulentos a ésta puedan hacer mal uso. La evaluación de este punto es uno de los más importantes en la interconexión de la Web con Bases de datos. A nivel de una red local, se puede permitir o impedir, a diferentes usuarios el acceso a cierta información, pero en la red mundial de Internet se necesita de controles mas efectivos en este sentido, ante posible espionaje, copia de datos, manipulación de éstos, etc. Para poder contrarrestar estos ataques se debe utilizar mecanismos que impidan cometer cualquier inserción no autorizada. La mayor parte de los sistemas utilizan uno o varios mecanismos de seguridad como encriptado, control de acceso, entre otras.

2.8.1 PROTOCOLOS DE SEGURIDAD

Como mecanismos de seguridad se utilizan estándares de seguridad implementados a través de protocolos. Entre los más difundidos tenemos:

- *SSL.- Secure Sockets Layer.-* Sistema que consiste en encriptar la información antes de enviarla, de forma que sólo el destinatario pueda conocer la información. Sólo funciona en interacción con otros exploradores compatibles. En la actualidad, la mayor parte de los exploradores web ofrecen soporte para SSL. Para que funcione el sistema SSL, es necesario tener una *ID digital (certificación de identidad)* emitido por un tercero. Este certificado permite identificar a alguien por otra persona acreditada para hacerlo y que confirma que es quien dice ser. Una vez en posesión de la certificación, se debe instalar y configurar en el servidor web.
- *.SET.- Secure Electronic Transaction.-* Su objetivo es proteger los datos sensibles de los compradores respetando la confidencialidad de los datos y autenticando la identidad de todas las partes que intervienen. Para esto, SET utiliza un sistema de firmas y certificados digitales que asegura que el emisor es quien dice ser y que sólo puede leer el mensaje el receptor autorizado.
- *MOSET.-* Una combinación de los dos anteriores, pues la primera parte de la transacción (el envío de información al comerciante) se hace empleando SSL, en tanto, la comunicación entre el comerciante y el banco utiliza SET.

2.8.2 PERMISOS DE ACCESO

La identificación del usuario es una de las formas de guardar la seguridad. Las identidades y permisos de usuarios están definidos en los Archivos de Control de Acceso. Pero la seguridad e integridad total de los datos puede conservarse, permitiendo el acceso a distintos campos de una base de datos,

solamente a usuarios autorizados para ello. En este sentido, los datos pueden ser presentados a través de la Web de una forma segura.

2.8.3 CONSIDERACIONES PARA LOS EJECUTABLES

Como regla general, sólo debe dar permiso de Ejecución a aquellos directorios virtuales que contengan aplicaciones CGI o ISAPI (API de Internet Server). A los directorios de contenido de World Wide Web sólo hay que asignarles permiso de Lectura. Estos son pasos necesarios para garantizar en gran medida la seguridad y confidencialidad de las transacciones, sin embargo, hay que estar consientes de que ningún mecanismo de seguridad será suficiente para contrarrestar los ataques mal intencionados.

CAPITULO III

MARCO HIPOTETICO

3.1 HIPOTESIS

Un adecuado planteamiento de directrices para la interoperabilidad de una base de datos distribuida, mejorará el rendimiento de un sistema de comercio electrónico.

3.2 OPERACIONALIZACION DE VARIBLES

HIPOTESIS	VARIABLES	CONCEPTO	INDICADORES
Un adecuado planteamiento de directrices para la interoperabilidad de una base de datos distribuida, mejorará el rendimiento de un sistema de comercio electrónico.	<p>Interoperabilidad de la Base de Datos Distribuida</p> <p>Tiempo de respuesta</p>	<p>Operación interna entre las instancias de la base de datos como: DBMS, catálogos, relaciones, fragmentos, registros y campos; respecto a requerimientos SQL y DML.</p> <p>Representa el intervalo de tiempo entre el envío de la solicitud por parte de la aplicación cliente y el retorno de su respuesta emitida por el servidor.</p>	<p>- Nivel de fragmentación</p> <p>- Frecuencia de acceso</p> <p>- Ubicación de los accesos</p> <p>- Volumen de información</p> <p>- Numero de solicitudes concurrentes</p> <p>- Volumen de cada solicitud</p> <p>- Ubicación de los solicitantes</p> <p>- Capacidad de la red</p>

Tabla 1. Operacionalización de variables

CAPITULO IV

ANALISIS DE LAS DIRECTRICES DE UNA BASE DE DATOS DISTRIBUIDA PARA LA OPTIMIZACION DE SISTEMAS DE COMERCIO ELECTRÓNICO

Los mayores problemas de rendimiento de las bases de datos distribuidas, se debe a que el proceso de distribución (fragmentación y asignación), se los realiza de forma errónea. Si bien es cierto en sistemas pequeños y simples un criterio no técnico puede dar buenos resultados, pero, cuando se trata de sistemas complejos, estos procedimientos empíricos, no son los adecuados.

La eficiencia del sistema de comercio electrónico, depende en la medida que la operación interna de la base de datos pueda satisfacer sus requerimientos. En tal virtud, en este capítulo se propone procedimientos, para que el proceso de fragmentación y asignación, sean realizados técnicamente, lo que asegura la máxima interoperabilidad de la base de datos distribuida, abocando en un incremento del rendimiento del sistema de comercio electrónico global.

4.1 ESQUEMA DEL PROCESO DE DISTRIBUCION

Para el análisis del proceso de distribución, se emplea el enfoque más utilizado actualmente, que es el enfoque en dos fases. Este esquema consiste en elaborar primeramente la fragmentación, para posteriormente proceder a la asignación de los fragmentos en los diferentes sitios.

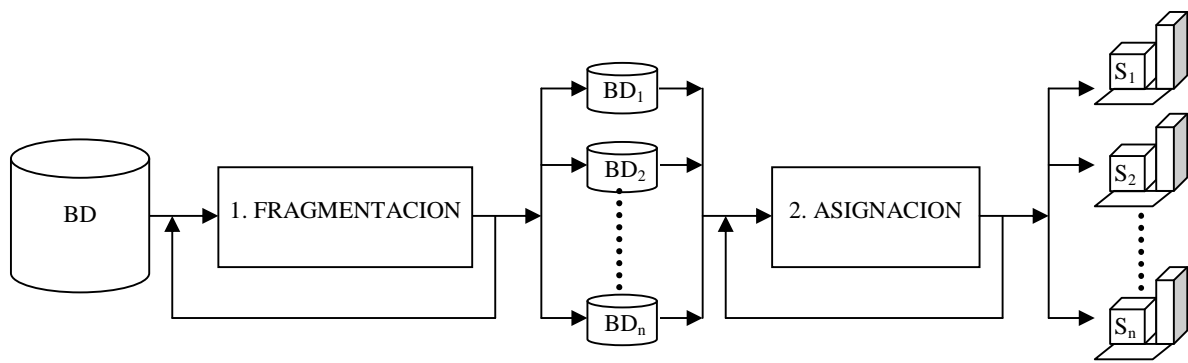


Fig. 14.- Esquema enfoque en dos fases

1. La fase de Fragmentación debe determinar fragmentos homogéneos de información desde el punto de vista del acceso, de modo que todos los fragmentos sean considerados uniformemente por las transacciones.
2. La fase de Asignación toma los fragmentos generados en la fase anterior y los asigna en los sitios más adecuados de acuerdo a las características de acceso de cada uno de los fragmentos y a las características de cada uno de los sitios.

4.2 REQUERIMIENTOS DE INFORMACIÓN

1. BASE DE DATOS

1. Esquema conceptual global de la base de datos: Diseño de la base de datos
2. Relaciones: R
3. Cardinalidad de cada relación: card (R).
4. Dependencia entre relaciones: Enlaces o Links

2. APLICACIONES

1. Aplicaciones que acceden a la base de datos: Programas Web
2. Consultas de usuario: Q
3. Predicados simples: P

4. Frecuencia de acceso de las consultas de usuario: $acc(q_i)$
5. Frecuencia de acceso de los predicados simples: $acc(p_i)$

4.3 PROCESO DE FRAGMENTACIÓN

El objetivo es conseguir la unidad apropiada de fragmentación, tal que cada uno de los fragmentos tenga un acceso uniforme y balanceado respecto al volumen y tiempo de respuesta que demandan las transacciones. El nivel de la fragmentación puede ir desde celdas, hasta relaciones.

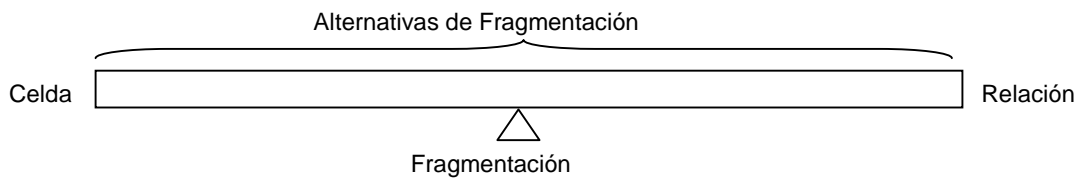


Fig. 15.- Esquema alternativas de fragmentación.

Al fragmentar la relación en un conjunto de sub-relaciones más pequeñas, las consultas hacen por lo tanto un menor acceso a las páginas de la base de datos.

4.3.1 FRAGMENTACION HORIZONTAL

La fragmentación horizontal divide a una relación R en un conjunto de sub-relaciones R_1, R_2, \dots, R_n , talque cada tupla de la relación R , se encuentra en algún fragmentos R_i .

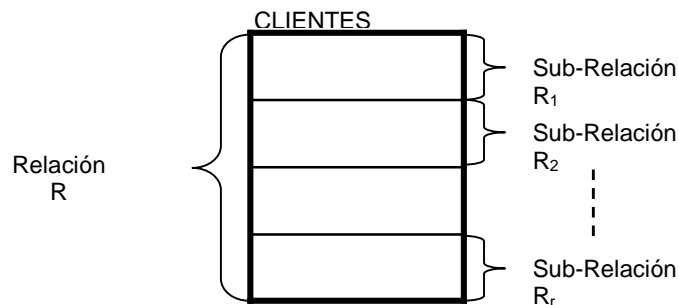


Fig. 16.- Esquema fragmentación horizontal- subrelaciones

La fragmentación horizontal se basa en el criterio de agrupar las tuplas que son frecuentemente accedidas en conjunto.

REPRESENTACION

Una relación R esta compuesta por un conjunto de atributos: $R(A_1, A_2, \dots, A_n)$

Cada atributo de la relación R, está definido en un dominio: $A_i: D_i$

Sobre la relación R se ejecuta un conjunto de predicados simples: $P = \{ p_1, p_2, \dots, p_m \}$

Un predicado simple tiene la forma:

$p_j: A_i \theta \text{ valor}$

$\theta \in \{ <, <=, =, <>, >=, > \}$

$\text{valor} \in D_i$

Ej. $p_1: \text{CLI_PROVINCIA} = \text{"Chimborazo"}$

$p_2: \text{CLI_CREDITO} >= 10.000$

Se puede definir dos variantes de la fragmentación horizontal: la fragmentación horizontal primaria y la fragmentación horizontal derivada. La fragmentación horizontal primaria se desarrolla empleando los predicados definidos en la misma relación (propietaria). Por el contrario, la fragmentación horizontal derivada consiste en dividir la relación (propietaria) a partir de los predicados definidos en otro (miembro).

4.3.1.1 FRAGMENTACION HORIZONTAL PRIMARIA

Un fragmento horizontal primario R_i deben contener todas las tuplas de R, que satisfagan el predicado minitérmino m_i , definido sobre la relación propietaria.

$$R_i = \sigma_{F_i} R$$

$$i = 1..n$$

F_i : fórmula de selección, predicado minitérmino m_i utilizado para obtener R_i .

El éxito de esta fragmentación horizontal primaria consiste en la adecuada agrupación de las tuplas, lo cual depende de la óptima definición del conjunto de predicados minitérmino M , a partir del cual se determina la forma y número de fragmentos horizontales adecuados de la relación R .

Forma: $R_i = \sigma_{F_i} R$

$i = 1..n$

F_i : fórmula de selección, predicado minitérmino m_i utilizado para obtener R_i .

Número: $\text{card}(M)$

M : conjunto de predicados minitérmino, $M = \{ m_1, m_2, \dots, m_r \}$

4.3.1.1.1 COMPLETITUD Y MINIMALIDAD DE LOS PREDICADOS SIMPLES

Es necesario que el conjunto de predicados simples P , sea completo y mínimo. Es decir, que P fragmente a una relación o fragmento en al menos dos partes, las cuales se acceden de forma diferente por al menos una consulta.

El conjunto de predicados simples P , es completo si y solo si la probabilidad de acceso de cada aplicación a cualquier par de tuplas del mismo fragmento es idéntica.

Ej. Si la relación **CLIENTES**, tiene definida la siguiente consulta:

q_1 : Listar los créditos de los clientes. Esta Consulta se ejecuta desde dos sitios, en el primero maneja a los clientes con créditos menores a 10.000 USD, en el segundo a los clientes con créditos superiores a 10.000 USD.

p_1 : $\text{CLI_CREDITO} < 10.000$

p_2 : $\text{CLI_CREDITO} \geq 10.000$

$P = \{ p_1, p_2 \}$

Entonces, P es completo, ya que cada tupla tiene la misma probabilidad de ser accesada (1/2).

El conjunto de predicados simples P, es mínimo, si todos sus predicados son relevantes. Un predicado es relevante cuando causa que a un fragmento f, se fragmente en f_i y f_j , y al menos una aplicación accede a f_i y a f_j de manera diferente.

Ej. El conjunto de predicados simple P, definidos anteriormente es mínimo, ya que causa que la relación CLIENTES (una relación es una generalización de un fragmento), se fragmente en:

$$\text{CLIENTES}_1: \sigma_{\text{CLI_CREDITO} < 10.000} \text{ CLIENTES}$$

$$\text{CLIENTES}_2: \sigma_{\text{CLI_CREDITO} \geq 10.000} \text{ CLIENTES}$$

Para este procedimiento se define un algoritmo, el cual genera un conjunto completo y mínimo de predicados simples P' a partir de una relación R y un conjunto de predicados simples P.



Fig. 17.- Esquema algoritmo de completitud y minimalidad

Este algoritmo requiere el cumplimiento de una regla elemental de los predicados.

Regla1: un predicado simple p_i debe particionar en al menos dos partes a un fragmento o relación, las cuales se accesan de forma diferente por al menos una consulta.

Algoritmo COM_MIN (R, P)

Entrada: Una relación R

Un conjunto de predicados simples P

Salida: Un conjunto completo y mínimo de predicados simples P'

1. Iniciación:

- a. Tomar un $p_i \in P$ tal que p_i particione a R de acuerdo con la Regla 1

- b. $P' = p_i$
- c. $P \leftarrow P - p_i$
- d. $F \leftarrow f_i$

2. Iterativamente analiza cada $p_i \in P$ para agregarlo a P'

- a. Tomar un $p_i \in P$ tal que p_i particione a f_k de acuerdo con la Regla 1
- b. $P' = P' \cup p_i$
- c. $P \leftarrow P - p_i$
- d. $F \leftarrow F \cup f_i$
- e. Si $\exists p_k \in P'$ el cual no es relevante, entonces

- i. $P' = P' - p_k$

- ii. $F \leftarrow F - f_k$

El algoritmo empieza encontrando un predicado que es relevante, y que particiona a la relación. Posteriormente, de manera iterativa agrega predicados a P' , asegurando minimalidad en cada paso. Al final de esto se obtiene un conjunto de predicados simples P' completo y mínimo. Este algoritmo es utilizado por el algoritmo de fraccionamiento horizontal primario, para obtener el conjunto de predicados minitérmino.

4.3.1.1.2 GENERACION DE PREDICADOS MINITÉRMINO

Para definir el conjunto de predicados minitérmino M , se analiza las combinaciones lógicas más utilizadas de los predicados simples p_i que se ejecutan sobre R , a esta conjunción de predicados simples p_i se los denomina predicados minitérmino m_i .

Al conjunto de predicados minitérmino M se lo puede representar como una expresión lógica en su forma normal conjuntiva.

$$M = \{ m_i \mid m_i = \bigwedge_{p_j \in P} p_j^* \}$$

$$i = 1..r$$

$$j = 1..m$$

$$p_j^* = p_j \circ p_j = \neg p_j^*$$

Ej. A partir de los dos predicados simples p_1 y p_2 :

$$p_1: \text{CLI_CREDITO} = \text{"Chimborazo"}$$

$$p_2: \text{CLI_CREDITO} \geq 10.000$$

Utilizando la representación conjuntiva se obtienen cuatro predicados minitérmino candidatos:

$$m_1: \text{CLI_PROVINCIA} = \text{"Chimborazo"} \wedge \text{CLI_CREDITO} \geq 10000$$

$$m_2: \text{CLI_PROVINCIA} = \text{"Chimborazo"} \wedge \text{NOT} (\text{CLI_CREDITO} \geq 10000)$$

$$m_3: \text{NOT} (\text{CLI_PROVINCIA} = \text{"Chimborazo"}) \wedge \text{CLI_CREDITO} \geq 10000$$

$$m_4: \text{NOT} (\text{CLI_PROVINCIA} = \text{"Chimborazo"}) \wedge \text{NOT} (\text{CLI_CREDITO} \geq 10000)$$

Como se puede apreciar, el establecimiento de los predicados minitérmino es trivial, pero, hay que tener en cuenta que el tamaño del conjunto de predicados minitérmino M , puede ser muy grande, de hecho, exponencial al número de predicados simples. Es necesario eliminar aquellos predicados minitérmino que resulten ser redundantes o contradictorios de acuerdo a un conjunto de implicaciones.

4.3.1.1.3 DEFINICIÓN DE IMPLICACIONES

Consiste en eliminar a través del análisis de un conjunto de implicaciones lógicas, los predicados minitérmino m_i que sean redundantes, contradictorios o inconsistentes.

Implicación de Igualdades

$$\text{Si } P = \{ p_1, p_2 \}$$

$$p_1: \text{atributo} = \text{valor1}$$

$$p_2: \text{atributo} = \text{valor2}$$

Dominio atributo = { valor1, valor2 }

Basándose en P, se puede definir los siguientes predicados minitérmino:

$m_1: (\text{atributo} = \text{valor1}) \wedge (\text{atributo} = \text{valor2})$

$m_2: (\text{atributo} = \text{valor1}) \wedge \neg (\text{atributo} = \text{valor2})$

$m_3: \neg (\text{atributo} = \text{valor1}) \wedge (\text{atributo} = \text{valor2})$

$m_4: \neg (\text{atributo} = \text{valor1}) \wedge \neg (\text{atributo} = \text{valor2})$

$M = \{ m_1, m_2, m_3, m_4 \}$

Implicaciones:

$i_1: (\text{atributo} = \text{valor1}) \rightarrow \neg (\text{atributo} = \text{valor2})$

$i_2: \neg (\text{atributo} = \text{valor1}) \rightarrow (\text{atributo} = \text{valor2})$

Se analiza los predicados minitérmino de acuerdo a las implicaciones i_1, i_2 :

PREDICADOS MINITÉRMINO (candidatos)

	m_1	m_2	m_3	m_4
IMPLICACIONES				
i_1	X			
i_2				X

Tabla 2. Predicados minitérminos

X: Indica que el predicado minitérmino m_i es inconsistente de acuerdo a la implicación i_j , por lo tanto quedan descartados, quedando el conjunto de predicados minitérmino así:

$M = \{ m_2, m_3 \}$

Implicación de Desigualdades

Si $P = \{ p_1, p_2 \}$

$p_1: \text{atributo} \leq \text{valor}$

p_2 : atributo > valor

Dominio atributo $\in D$, D es cualquier dominio que tiene una representación numérica

Basándose en P, se puede definir los siguientes predicados minitermino:

m_1 : (atributo \leq valor) \wedge (atributo > valor)

m_2 : (atributo \leq valor) $\wedge \neg$ (atributo > valor)

m_3 : \neg (atributo \leq valor) \wedge (atributo > valor)

m_4 : \neg (atributo \leq valor) $\wedge \neg$ (atributo > valor)

Implicaciones:

i_1 : (atributo \leq valor) $\rightarrow \neg$ (atributo > valor)

i_2 : \neg (atributo \leq valor) \rightarrow (atributo > valor)

i_3 : (atributo > valor) $\rightarrow \neg$ (atributo \leq valor)

i_4 : \neg (atributo > valor) \rightarrow (atributo \leq valor)

Se analiza los predicados minitermino de acuerdo a las implicaciones i_1, i_2, i_3, i_4 :

PREDICADOS MINITÉRMINO (candidatos)

	m_1	m_2	m_3	m_4
i_1	X			
i_2				X
i_3				
i_4				

Tabla 3. Predicados miniterminos- implicaciones

X: Indica que el predicado minitermino m_i es inconsistente de acuerdo a la implicación i_j , por lo tanto quedan descartados, quedando el conjunto de predicados minitermino así:

$$M = \{ m_2, m_3 \}$$

Al final de la eliminación de los predicados minitermino redundantes, contradictorios o inconsistentes de acuerdo a las implicaciones, se obtiene el conjunto de predicados minitermino definitivo

$$M = \{ m_1, m_2, \dots, m_r \}$$

los cuales definen la manera en que la relación R es fragmentada.

$$F_R = \{ R_1, R_2, \dots, R_r \}$$

4.3.1.1.4 ALGORITMO PARA LA FRAGMENTACION HORIZONTAL PRIMARIA

Todos los procesos anteriores pueden resumirse en un algoritmo que se presenta a continuación:

Entrada: Una relación R

Un conjunto de predicados simples P

Salida: Un conjunto de predicados minitermino M, de acuerdo al cual la relación R es fragmentada

1. Determinar el conjunto P' de predicados simples completo y mínimo: $P' \leftarrow \text{COM_MIN} (R, P)$
2. Determinar el conjunto M de predicados minitermino
3. Determinar el conjunto I de implicaciones, entre los $p_i \in P$
4. Eliminar de M los predicados contradictorios respecto a I

4.3.1.2 FRAGMENTACION HORIZONTAL DERIVADA

Una fragmentación horizontal derivada se define sobre una relación miembro de acuerdo a la fragmentación primaria de la relación propietaria. El enlace entre las relaciones miembro y propietaria se define mediante un semi-join. El semi-join de una relación R, definida sobre el conjunto de atributos A, con otra relación S, definida sobre un conjunto de atributos B, es el subconjunto de tuplas de R que participan en el join con S.

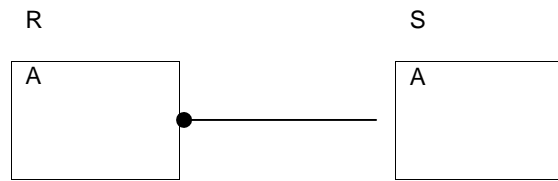


Fig. 18.- Esquema semi-join de una relación

Dado un enlace entre dos relaciones, donde S es la relación propietaria y R la relación miembro, la fragmentación horizontal derivada de R se define como:

$$R_i = R \lt S_i$$

$$i = 1..w$$

w: es el número máximo de fragmentos definidos sobre R.

S_i : Un fragmento generado a partir de un proceso de fragmentación horizontal primaria en S

$$S_i = \sigma_{F_i} S$$

F_i : fórmula del predicado minitérmino de acuerdo a la cual la fragmentación horizontal primaria sobre la relación propietaria S se lleva a cabo

En un esquema de base de datos por lo general una relación tiene más de un enlace, por tal motivo hay dos posibilidades de efectuar la fragmentación horizontal derivada.

1. Fragmentación con mejores características de join.- Para beneficiar a la consulta que haga uso de las dos relaciones.
2. Fragmentación utilizada en más aplicaciones.- Para facilitar el acceso a los usuarios que hagan mayor uso de los datos.

Para la decisión de cual de las alternativas seleccionar se debe analizar la que menor impacto represente en el rendimiento del sistema.

4.3.2 FRAGMENTACION VERTICAL

La fragmentación vertical divide a una relación R en un conjunto de sub-relaciones R_1, R_2, \dots, R_n , tal que cada atributo de la relación R, se encuentra en algún fragmentos R_i ,

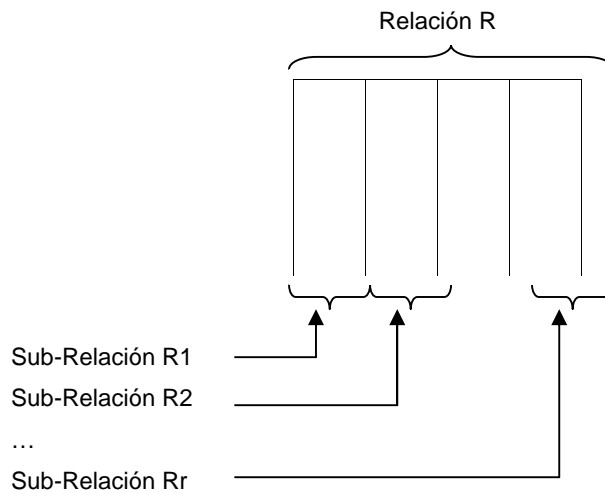


Fig. 19.- Esquema fragmentación vertical - subrelación

La fragmentación vertical se basa en el criterio de agrupar los atributos que son frecuentemente accedados en conjunto. Una característica de la fragmentación vertical es que la clave de la relación se replica en todos los fragmentos, a través de la cual se realiza la reconstrucción de la relación global.

REPRESENTACION

Una relación R esta compuesta por un conjunto de atributos: $R(A_1, A_2, \dots, A_n)$

Sobre la relación R se ejecuta un conjunto de consultas: $Q = \{ q_1, q_2, \dots, q_q \}$

Las consultas q_i son ejecutadas desde varios sitios: $S = \{ s_1, s_2, \dots, s_s \}$

El éxito de la fragmentación vertical consiste en la adecuada agrupación de los atributos. Lo cual depende de la óptima definición de la afinidad entre los atributos de la relación R, a partir del cual se determina la forma y número de fragmentos horizontales adecuados de la relación R.

Forma: $R_i = \Pi_A R$

$$i = 1..n$$

A: atributos de la relación R utilizado para obtener R_i

Número: n: Número de grupos de afinidad obtenido

4.3.2.1 DEFINICION DE USO DE ATRIBUTOS

Determina que consultas acceden a que atributos. Para esto se puede elaborar una Matriz de Uso de Atributos A_j , respecto a las consultas q_i , donde

$$\text{use}(q_i, A_j) = \begin{cases} 1 & \text{Si el atributo } A_j \text{ es referido por la consulta } q_i \\ 0 & \text{en caso contrario} \end{cases}$$

$\text{use}(q_i, A_j)$: es el valor de un elemento en la matriz de uso, para eso se debe conocer los atributos a los que acceden cada una de las consultas en las aplicaciones.

	A_1	A_2		A_n
q_1	1	0		1
q_2	0	1		1
q_q	1	1		0

Tab. 4.- Matriz de Uso de Atributos

A partir de la matriz de uso de atributos se puede definir las consultas q_k que acceden a A_i y A_j como:

$$q_k(A_i, A_j) = \text{use}(q_k, A_i) \wedge \text{use}(q_k, A_j)$$

4.3.2.2 FRECUENCIA DE ACCESO POR SITIO

Se debe conocer la frecuencia de ejecución de cada consulta en cada uno de los sitios.

	S₁	S₂		S_n
q₁	10	15		5
q₂	20	0		15
q_q	5	15		10

Tab. 5.- Matriz de Acceso por Sitios

A partir de la matriz de acceso por sitios se define el número de accesos a la consulta q_k en cada sitio S_i como:

$$\text{acc}_i(q_k) = \# \text{ de accesos a la consulta } q_k(A_i, A_j) \text{ en el sitio } S_i$$

4.3.2.3 DEFINICION DE AFINIDAD ENTRE ATRIBUTOS

La afinidad determina el grado en que están relacionados los atributos de una relación R, respecto a un conjunto de consultas Q. Este valor se determina a partir de la matriz de uso de atributos (Tab. 4) y la matriz de acceso por sitios (Tab, 5). La afinidad entre los atributos de una relación se determina como:

$$\text{aff}(A_i, A_j) = \sum_{\forall q_k(A_i, A_j)} \sum_{\forall S_i} \text{acc}_i(q_k)$$

- $q_k(A_i, A_j)$: consultas q_k que accesan a A_i y A_j .
- $\text{acc}_i(q_k)$: frecuencia de acceso a la consulta q_k en el sitio S_i

A partir de esta ecuación se define la matriz de afinidad de atributos AA

	A₁	A₂		A_n
A₁	60	30		30
A₂	30	35		65
A_n	30	65		65

Tab. 6- Matriz de Afinidad entre Atributos

Una característica de la matriz de afinidad (AA) entre atributos es su simetría.

4.3.2.4 AGRUPACIÓN ENTRE ATRIBUTOS

Se debe agrupar los atributos de una relación basándose en los valores de afinidad entre los mismos, de manera de maximizar la medida global de afinidad, dada por:

$$AM = \sum_{i=1}^n \sum_{j=1}^n aff(A_i, A_j) [aff(A_i, A_{j-1}) + aff(A_i, A_{j+1}) + aff(A_{i-1}, A_j) + aff(A_{i+1}, A_j)]$$

donde: $aff(A_0, A_j) = aff(A_i, A_0) = aff(A_{n+1}, A_j) = aff(A_i, A_{n+1}) = 0$

Como se indicó anteriormente la matriz de afinidades AA es simétrica, la función anterior se reduce a:

$$AM = \sum_{i=1}^n \sum_{j=1}^n aff(A_i, A_j) [aff(A_i, A_{j-1}) + aff(A_i, A_{j+1})]$$

Para maximizar esta función se utiliza un algoritmo que a partir de la matriz de afinidad reorganiza el orden de los atributos, formando grupos, donde los atributos dentro de cada grupo presentan la mas alta afinidad uno con otro.

4.3.2.5 ALGORITMO DE AGRUPACIÓN

Entrada: Matriz de afinidad de atributos: AA

Salida: Matriz de afinidad de agrupación: CA

1. Inicialización: Fijar una columna de AA y ponerla en CA
2. Iteración: Colocar las n-i columnas restantes en las I+1 posiciones restantes de la matriz CA. Para cada columna elegir la ubicación que causa mayor contribución en la medida global de afinidad.
3. Ordenar filas: Ordenar las filas de acuerdo al orden de las columnas

Para el segundo paso, la contribución se define como:

$$\text{cont}(A_i, A_k, A_j) = 2 \text{ bond}(A_i, A_k) + 2 \text{ bond}(A_k, A_j) - 2 \text{ bond}(A_i, A_j)$$

donde:

$$\text{bond}(A_x, A_y) = \sum_{z=1}^n \text{aff}(A_z, A_x) \text{aff}(A_z, A_y)$$

Una vez concluido este proceso se tiene una matriz agrupada de atributos CA ordenada de acuerdo a la afinidad de los mismos.

	A ₁	A _n		A ₂
A ₁	60	30		30
A _n	30	65		65
A ₂	30	65		35

Tab. 7- Matriz agrupada de Atributos

4.3.2.6 ALGORITMO DE FRAGMENTACION VERTICAL

A partir de la matriz agrupada de atributos CA, el objetivo es determinar los atributos que son accedados en conjunto. Un conjunto es $\{A_1, \dots, A_i\}$ y un segundo conjunto es $\{A_{i+1}, \dots, A_n\}$ que representan la esquina superior izquierda TA y la esquina inferior derecha BA respectivamente de la matriz CA.

	A ₁	A _i	A _{i+1}	A _n
A ₁	TA			
A _i				
A _{i+1}			BA	
A _n				

Tab. 8.- Matriz accedadas en conjunto

Sea $Q = \{q_1, q_2, \dots, q_q\}$, hay que definir el conjunto de consultas que accedan a TA, BA, o ambas.

TQ = conjunto de aplicaciones que accesan únicamente a TA

BQ = conjunto de aplicaciones que accesan únicamente a BA

OQ = conjunto de aplicaciones que accesan tanto a TA como a BA

CTQ = número total de accesos a atributos por aplicaciones que accesan únicamente a TA

CBQ = número total de accesos a atributos por aplicaciones que accesan únicamente a BA

COQ = número total de accesos a atributos por aplicaciones que accesan tanto a TA como BA

El problema es encontrar el punto a lo largo de la diagonal que maximiza la función objetivo.

$$z = CTQ * CBQ - COQ^2$$

Entrada: Matriz CA, relación R, matriz $ref(q_i)$, matriz de $acc(q_i)$

Salida: Conjunto de fragmentos (F)

Inicio

calcular CTQ_{n-1}

calcular CBQ_{n-1}

calcular COQ_{n-1}

$$z_{max} = CTQ_{n-1} * CBQ_{n-1} - COQ_{n-1}^2$$

para $i = n-2$ bajando hasta 1 hacer

calcular CTQ_i

calcular CBQ_i

calcular COQ_i

$$z = CTQ_i * CBQ_i - COQ_i^2$$

si $z > z_{max}$ entonces $z_{max} = z$

TA = atributos más izquierda de z

BA = atributos más derecha de z

$$R_1 = \Pi_{TA} RUK$$

$$R_2 = \Pi_{TB} RUK$$

$$F = \{R_1, R_2\}$$

Fin

4.3.3 FRAGMENTACION MIXTA

Para conseguir una fragmentación mixta, se debe aplicar la fragmentación vertical seguida de una horizontal o viceversa. Las reglas que se aplican a cada una de ellas, se mantienen vigentes durante el proceso de fragmentación mixta. Aunque se puede definir un número finito de niveles al generar el árbol de particionamiento estructurado, en la práctica se recomienda que no haya más de dos niveles por dos razones:

1. El proceso de normalización aplicado en el diseño ya produce las relaciones adecuadas
2. Los beneficios del rendimiento pueden desperdiciarse cuando se requieren muchas operaciones para reconstruir la relación original.

4.4 PROCESO DE ASIGNACION

Con el conjunto de fragmentos $F = \{F_1, F_2, \dots, F_n\}$, las consultas $Q = \{Q_1, Q_2, \dots, Q_q\}$ que operan sobre dichos fragmentos y un conjunto de sitios $S = \{S_1, S_2, \dots, S_n\}$ disponibles para su utilización, se debe analizar cual de las alternativas de asignación proporciona mayor disponibilidad, incremento del paralelismo, menor tiempo de recuperación

4.4.1 INFORMACION NECESARIA

- Base de datos
 - Selectividad de c/fragmento: $sel(F_j)$
 - Tamaño de c/fragmento: $size(F_j)$

	F_1	F_2	...	F_n
$sel(F_j)$				
$size(F_j)$				

Tab. 9- Matriz de selectividad

- Aplicaciones

- # accesos de lectura de q_i a F_j : RR_{ij}
- # acceso de actualización de q_i a F_j : UR_{ij}
- Vector O de valores $O(i)$, que es la localidad que da origen a la consulta
- Matrices UM y RM de elementos u_{ij} y r_{ij} :

$$u_{ij} = \begin{cases} 1 & \text{si } q_i \text{ actualiza } F_j \\ 0 & \text{en otro caso} \end{cases} \quad r_{ij} = \begin{cases} 1 & \text{si } q_i \text{ lee } F_j \\ 0 & \text{en otro caso} \end{cases}$$

RR/UR	F_1	F_2	...	F_n		S_1	S_2	...	S_m	RR/UR	F_1	F_2	...	F_n
Q_1					$O(i)$					Q_1				
Q_2										Q_2				
...										...				
Q_q										Q_q				

- Localidades

- Unidad de coste de almacenamiento: UCA_k
- Unidad de coste de proceso: UCP_k

	S_1	S_2	...	S_m
UCA_k				
UCP_k				

- Red

- Coste de comunicación entre dos localidades S_i y S_j : g_{ij}
- Tamaño de la trama: f_{size}

Q_{ij}	S_1	S_2	...	S_m
S_1				
S_2				
...				
S_m				

Tab. 10- Matrices UM y RM de elementos u_{ij} y r_{ij}

4.4.2 OPTIMIZACION DE LOS COSTOS DE LA ASIGNACION

Se debe definir la distribución óptima de F en S de acuerdo a dos criterios:

1. Costo mínimo.- Hay que minimizar el costo combinado de:
 - a. Almacenamiento: escritura de los fragmentos
 - b. Procesamiento: lectura, escritura y actualización de cada fragmento
 - c. Comunicación: envío de solicitudes y respuestas
2. Rendimiento.- Se considera las métricas de:
 - a. Tiempo de respuesta: intervalo entre el envío de la solicitud y llegada de su respuesta.
 - b. Throughput: número de trabajos procesados en una unidad de tiempo

Las restricciones a tomar en cuenta son las capacidades de cada nodo para dichas tareas. Los factores de costo de almacenamiento, procesamiento y comunicación descritos anteriormente los puede representar en un costo total como:

$$C_t = \sum_{\forall q_i \in Q} CP_{q_i} + \sum_{\forall S_i \in S} \sum_{\forall F_j \in F} CA_{jk}$$

El proceso de optimización consiste en encontrar el mínimo valor para la función anterior. A continuación analizaremos cada uno de los elementos de esta ecuación

CP_{q_i} es el coste de procesamiento de la consulta q_i

$$CP_{q_i} = CP_i + CT_i$$

CA_{jk} es el costo de almacenamiento del fragmento F_j en el sitio S_k

$$CA_{jk} = UCA_k * \text{size}(F_j) * x_{jk}$$

$$x_{jk} = \begin{cases} 1 & \text{si } F_j \text{ está almacenado en } S_k \\ 0 & \text{en otro caso} \end{cases}$$

4.4.3 CONSIDERACIONES EN LAS ALTERNATIVAS DE ASIGNACIÓN

Cuando una serie de fragmentos se asignan, éstos pueden replicarse para mantener una copia o varias idénticas. Cada copia se almacena en una localidad diferente, lo que resulta en una repetición de la información. En tal virtud es conveniente analizar las ventajas y desventajas de esta.

4.4.3.1 MAYOR DISPONIBILIDAD

Si falla una de las localidades que contienen la relación r , puede disponerse de ésta en otra localidad. Así, el sistema puede continuar procesando consultas que impliquen a r a pesar de haber fallado una localidad. Por lo tanto, todo esto gira en torno a la seguridad y a la eficiencia de las consultas de lectura.

4.4.3.2 INCREMENTO DEL PARALELISMO

En el caso en que la mayor parte de los accesos a la relación r resulten sólo en la lectura de la relación, varias localidades podrán procesar consultas que involucren a r en paralelo. Mientras más copias de r existan, mayor será la probabilidad de que los datos requeridos se encuentren en la localidad donde se está ejecutando la transacción. Por tanto, la replica de los datos reduce al mínimo el movimiento de información entre localidades. Así, un buen parámetro para afrontar el grado de réplica consistiría en sopesar la cantidad de consultas de lectura que se efectuarán, así como el número de operaciones de escritura que se llevarán a cabo. En una red donde las consultas que se procesen sean mayoritariamente de lectura, se podría alcanzar un alto grado de réplica, no así en el caso contrario.

4.4.3.3 MAYOR TIEMPO EXTRA EN ACTUALIZACIONES

El sistema debe asegurarse de que todas las copias de la relación r sean consistentes, pues de otra manera pueden hacerse cálculos erróneos. Esto implica que cada vez que se actualice r , la actualización debe propagarse a todas las localidades que contengan copias, lo que resulta en un mayor tiempo extra. Por ejemplo, en un sistema bancario, donde se repite la información de cuentas en varias

localidades, es necesario que las transacciones garanticen que el saldo de una cuenta determinada sea el mismo en todas las localidades.

4.4.3.4 RECUPERACIÓN

Una vez que un sitio, servicio, aplicación u otro elemento haya sido levantado es necesario recurrir al proceso de sincronización de información. Aunque en la actualidad existen herramientas automáticas de recuperación, estas siempre deberán estar supervisadas por el administrador de la base de datos.

En la actualidad existen diferentes tipos de herramientas para implantar una base de datos en la World Wide Web. La capacidad y alcance del software disponible depende directamente del:

- hardware con que se cuente para la implantación de la base de datos,
- sistema operativo en función,
- rango de usuarios contemplado,
- el crecimiento a futuro de la base de datos ya que del tamaño de las tablas y/o fragmentos dependerá también la rapidez de consulta,
- la calidad de los dispositivos de comunicación y la capacidad del servidor tienen gran relevancia para ofrecer un buen servicio a los usuarios.

4.5 FACTORES ADICIONALES A CONSIDERAR EN EL RENDIMIENTO DE UN BDD

4.5.1 AFINAMIENTO DEL SISTEMA OPERATIVO

Sin duda alguna uno de los factores más importantes para un buen desempeño en el rendimiento, es el sistema operativo. Esto determina la posible potencia de la base de datos. Los sistemas operativos han desarrollado bases sólidas para este tipo de aplicaciones. Existen en el mercado sistemas UNIX, Windows, OS/2 con grandes capacidades y con un alto grado de estabilidad.

En los últimos años se han desarrollado *sistemas operativos con capacidades específicas de interoperabilidad de bases de datos*, con herramientas de comunicación, como por ejemplo:

- Las máquinas UNIX que corren SunOS, Solaris, HP-UX o cualquier otra variedad de UNIX son las plataformas preferidas para bases de datos de gran tamaño.
- La aceptación que Windows NT ha tenido en el mercado de los sistemas operativos de red y la madurez que ha alcanzando en los últimos Resource Kit proporcionan también una alternativa atractiva.
- Linux que se ha convertido en el sistema operativo para PC con capacidades Internet, pues configura todos los servicios en una computadora sin exigir gran requerimiento de hardware, aunque este le falta un gran camino por recorrer para poder estar al nivel de los dos anteriores, sin embargo, no les pide nada cuando de bases de datos pequeñas y medianas se trata.

En todo caso, en cualquiera de los sistemas operativos que se elija, un elemento importante de considerar es el manejo de la “*caché*”. En las aplicaciones cliente-servidor es importante minimizar las interacciones entre un cliente y el servidor, es por eso que las máquinas se valen del protocolo ARP de TCP/IP para mantener una memoria intermedia (caché) de respuestas para mejorar la eficiencia de las búsquedas redundantes de información que surjan posteriormente. El proceso de memoria intermedia llamado “*caching*” mejora el desempeño del modelo cliente-servidor. Un ejemplo real de esta aplicación se da en el caché del browser utilizado para navegar, en donde se alojan todos los gráficos y de más archivos de las páginas que han sido visitadas, de forma que si en una sesión se vuelve a visitar el sitio no es necesario que el servidor vuelva a enviar esa información pues esta se tiene en el caché y puede ser recuperada de forma inmediata.

4.5.2 OPTIMIZACION EN LA TECNOLOGÍA DE ALMACENAMIENTO

Al diseñar una base de datos se debe considerar no solo el tipo de dato, si no la cantidad de datos que en ella se va a almacenar. Si la cantidad de registros está en la escala de los cientos de millones, el espacio de almacenamiento así como su respaldo y disponibilidad se vuelve crucial. En tal virtud, es necesario definir mecanismos que mantenga a dicha información en niveles adecuados de calidad (fiabilidad, disponibilidad, etc.), de acuerdo a las características propias del sistema de comercio electrónico.

4.5.3 ALMACENAMIENTO PRIMARIO

Para asegurar la disponibilidad de la información se deben considerar los controladores nativos del disco duro, una buena alternativa son los controladores *SCSI (Small Computer Systemes Interface)* tanto para el almacenamiento primario como para el reflejo de datos. Los controladores SCSI son la norma en equipos Macintosh y UNIX. Estos pueden manejar hasta siete dispositivos cada uno y es posible utilizar varios controladores simultáneamente. Es un interfaz de bus de 8 bits que permite que dos dispositivos se comuniquen a la vez, esto es de principal a periféricos o de periférico a periférico.

SCSI (SCSI-1).- Proporciona una tasa de transferencia de hasta 5MBytes y puede conectar varios periféricos ocupando solamente una ranura de expansión en la computadora.

SCSI-2.- Proporciona colas de órdenes y una opción síncrona "Fast SCSI" que proporciona una tasa de transferencia de 10Mbps (8 bits), con un cable "B" secundario para datos de 16 y 32 bits. "Wide SCSI" proporciona hasta una tasa de transferencia de 40Mbps con un cable "P" propuesto de 68 conductores.

SCSI-3.- Propone características realzadas y capacidad de manejar más de ocho dispositivos

Serial ATA.- Es una tecnología de almacenamiento que trabaja con una comunicación serial, a una tasa de transferencia de 150Mbps.

Analizando se puede ver que este tipo de dispositivos agiliza el acceso y recuperación de datos ya que trabajan con una tasa de 5, 10, 40, 150 Mbps a diferencia de los discos convencionales que trabajan a (5400 rpm o 7200 rpm), una tasa de transferencia de 3.88 Mbps, lo que quiere decir que en el acceso a la información se puede tener un incremento proporcional en el rendimiento de 1.29%, 2.57%, 10.31%, 38.66%.

4.5.4 RESPALDO

En la gran mayoría de sistemas, la información que se maneja es crítica y crucial para la toma de decisiones y como ventaja competitiva frente a las demás empresas, por tal motivo su respaldo es imprescindible. Lógicamente esta obligatoriedad de mantener uno o varios respaldos de la información es costosa si se tratan de volúmenes considerables. No obstante sus beneficios son incuantificables cuando se ve en la necesidad de recurrir a ellos. Los costos que se incurren en un sistema de respaldo van proporcionalmente al grado de fiabilidad de los mismos. Una manera de poder definirlos puede ser:

$$Costo_respaldo_total = \sum_1^n Costo_respaldo(i)$$

Donde n es el número o cantidad de respaldos que de la base de datos se ha sacado y cada uno de los cuales representa entre otros los siguientes costos:

$$Costo_respaldo(i) = procesamiento + transmisión + almacenamiento + indisponibilidad_BD$$

Cada uno de los factores (sumandos) inevitablemente estará presente cada vez que se haga un Back Up de la información.

Procesamiento.- Es el tiempo necesario para ejecutar el proceso de respaldo, esto incluye principalmente tiempo de procesador y memoria.

Transmisión.- Es el tiempo que se necesita para transmitir la base de datos a una unidad o dispositivo de respaldo.

Almacenamiento.- Es el costo que implica mantener la información en un determinado dispositivo de almacenamiento. Ej. Servidor de respaldos, tape back-up, etc.

Indisponibilidad_BD.- Es el costo que representa tener en inactividad el site mientras los procesos anteriores se lleva a cabo.

4.5.5 ELECCIÓN DE LA TECNOLOGÍA RAID

Hoy en día los métodos más difundidos para el resguardo de información utilizan la tecnología *Raid* (*Redundant Arrays of Inexpensive Disks*) con dispositivos SCSI. Es un agrupamiento de discos en la que los datos se copian en varias unidades, proporcionando un caudal más rápido, tolerancia de fallos (espejos) y corrección de errores mediante un mecanismo de paridad.

En este tipo de tecnología se maneja diferentes niveles. El nivel 3 se utiliza para grandes transferencias de bloques como imágenes, comunicación en tiempo real, alimentación satélite, etc. El nivel 5 es el más común actualmente, por lo que haremos una breve revisión de la misma.

Raid 5.- En este tipo de tecnología los datos y la paridad son guardados en los mismos discos, por lo que se consigue aumentar la velocidad de demanda, ya que cada disco puede satisfacer una demanda independientemente de los demás. También guarda la paridad del dato dentro de los discos y no hace falta un disco para guardar dichas paridades.

La paridad se genera haciendo un XOR de los datos A0, B0, C0, D0 creando la zona de paridad PAR0.

La paridad nunca se guarda en los discos que contienen los datos que han generado dicha paridad, ya

que en el caso en que uno de ellos se estropeará como por ejemplo el dato A0, bastaría con regenerar la banda B0, C0, D0, PAR0 para que el dato vuelva a reestablecerse.

4.6 OPTIMIZACION DEL PROCESAMIENTO DE CONSULTAS

La localización requiere una optimización en operaciones globales que son tomadas como una parte de la consulta global optimizada. La optimización de consultas globales involucra la permutación del orden de operaciones en una consulta, determinando la ejecución de varias operaciones distribuidas e identificando el mejor algoritmo de ejecución para las operaciones distribuidas.

La optimización de consultas paralelas toma ventaja del paralelismo intra operación y del paralelismo Inter operación.

- *Paralelismo intra operación.*- Esta consiste en ejecutar una operación en varios nodos de una máquina con varios procesadores.
- *Paralelismo Inter operación.*- Ocurre cuando dos o más operaciones son ejecutadas en paralelo, sea utilizando pipeline o independientemente.
- *Paralelismo Independiente.*- Se da cuando ocurren las operaciones al mismo tiempo o en orden arbitrario. El paralelismo independiente es posible solo cuando las operaciones no involucran los mismos datos.

4.7 AFINAMIENTO DEL CONECTOR DE BASE DE DATOS INTERNET (IDC)

Uno de los factores cruciales en la calidad de rendimiento en la interoperabilidad de la base de datos con el sistema de comercio electrónico sin duda es la configuración del Conector de Base de Datos Internet. Entre los parámetros que hay que darles principal atención son aquellos cuya configuración

requieren de conocimiento, información estadística y pruebas de afinamiento par encontrar el mejor rendimiento. Entre los campos susceptibles de esta operación tenemos:

- Campos opcionales de los archivos .idc
- Campos opcionales avanzados de ODBC

En el Anexo IV se describe el contenido significado de cada uno de los parámetros que se pueden configuran en dichos archivos.

4.8 OPTIMIZACION DE CONSULTAS

En el proceso de optimización es necesario tomar en cuenta todos los aspectos que afectan el buen desempeño de la Base de Datos. Una tarea que se efectúa ya sea con diseño centralizado o distribuido, es la aplicación de las técnicas de optimización de consultas, las mismas que definen la forma de estructurar los predicados de las sentencias SQL con el fin de minimizar los accesos y operaciones que se llevan a cabo durante la ejecución de las sentencias.

Un aspecto adicional a considerar en la búsqueda de eficiencia en las transacciones es la optimización de consultas por agrupación de las mismas. En un archivo .idc, se puede conseguir esto, ya que es posible agrupar consultas SQL de dos formas distintas:

1. como consultas por lotes, o
2. como consultas múltiples.

4.8.1 POR AGRUPACIÓN DE CONSULTAS POR LOTES

Para poder llevar a cabo este proceso es necesario que la base de datos a utilizar incluya características que permita procesar simultáneamente varias consultas en una única instrucción SQL. En este caso se deben dar a las instrucciones el formato por lotes con el fin de optimizar el rendimiento. Por ejemplo una sentencia agrupada por lotes podría ser así:

SQLStatement:

```
+INSERT into facturas(f_numero, f_fecha, f_valor, c_numero) values (%numero%, getdate(),  
%valor%, %ncliente%)
```

```
+SELECT f_numero, f_fecha, f_valor from facturas where c_numero = %ncliente%
```

```
+SELECT count(*), sum(f_valor) from facturas where c_numero = %ncliente%
```

4.8.2 POR AGRUPACIÓN DE CONSULTAS MULTIPLES

En el caso que la base de datos a utilizar no pueda procesar una serie de consultas SQL simultáneamente, es decir por lotes, se puede optar por formular consultas como consultas múltiples.

Por ejemplo:

SQLStatement:

```
+INSERT into facturas(f_numero, f_fecha, f_valor, c_numero) values (%numero%, getdate(),  
%valor%, %ncliente%)
```

SQLStatement:

```
+SELECT f_numero, f_fecha, f_valor from facturas where c_numero = %ncliente%
```

SQLStatement:

```
+SELECT count(*), sum(f_valor) from facturas where c_numero = %ncliente%
```

La diferencia entre una u otra agrupación es que las consultas por lotes se procesan juntas inmediatamente, mientras que las consultas múltiples se procesan una a una. En tal virtud, *si la Base de datos puede administrar consultas por lotes, es recomendable que las mismas se agrupen de esta forma ya que así se obtendrá mejor rendimiento.*

4.9 CONSIDERACIONES EN LOS ENTORNOS DISTRIBUIDOS

En los entornos distribuidos la probabilidad de fallos aumenta por la complejidad inherente de sistemas de esta naturaleza. Es necesario identificar estos problemas así como las tácticas para solucionar dichos inconvenientes, ya que las inconsistencias que puedan ocurrir afectan la operabilidad que se pretende conseguir.

4.9.1 PROBLEMAS DE LOS ENTORNOS DISTRIBUIDOS

Fallo de los nodos.- Cuando un nodo falla, el sistema deberá continuar trabajando con los nodos que aún funcionan. Si el nodo a recuperar es una base de datos local, se deberán separar los datos entre los nodos restantes antes de volver a unir de nuevo el sistema.

Copias múltiples de fragmentos de datos.- El subsistema encargado del control de concurrencia es el responsable de mantener la consistencia en todas las copias que se realicen y el subsistema que realiza la recuperación es el responsable de hacer copias consistentes de los datos de los nodos que han fallado y que después se recuperarán.

Transacción distribuida correcta.- Se pueden producir fallos durante la ejecución de una transacción correcta si se plantea el caso de que al acceder a alguno de los nodos que intervienen en la transacción, dicho nodo falla.

Fallo de las conexiones de comunicaciones.- El sistema debe ser capaz de tratar los posibles fallos que se produzcan en las comunicaciones entre nodos. El caso más extremo es el que se produce cuando se divide la red. Esto puede producir la separación de dos o más particiones donde las particiones de cada nodo pueden comunicarse entre sí pero no con particiones de otros nodos.

4.9.2 SUBSISTEMAS EN LOS ENTORNOS DISTRIBUIDOS

Para implementar las soluciones a estos problemas, debemos analizar cada uno de los sitios que son parte del sistema distribuido. Este enfoque consiste en optimizar cada uno de los nodos en particular, para así optimizar el rendimiento global. En tal virtud supondremos que los datos se encuentran almacenados en un único nodo sin repetición. De ésta manera sólo existirá un único catálogo y un único Data Manager (DM) encargados del control y acceso a las distintas partes de los datos. En este contexto para mantener la consistencia de los datos en el entorno distribuido contaremos con los siguientes elementos que podemos afinar para obtener mejores resultados:

Catálogo (scheduler).- Es uno o varios programas encargados de controlar la ejecución concurrente de las transacciones.

CM (Cache Manager).- Es un subsistema que se encarga de mover los datos entre las memorias volátiles y no volátiles, en respuesta a las peticiones de los niveles más altos del sistema de bases de datos. Sus operaciones son Fetch(x) y Flush(x).

RM (Recovery Manager).- Es un subsistema que asegura que la base de datos contenga los efectos de la ejecución de transacciones correctas y ninguno de incorrectas. Sus operaciones son Start, Commit, Abort, Read, Write, que utilizan a su vez los servicios del CM.

DM (Data Manager).- Unifica las llamadas a los servicios del CM y el RM.

TM (Transaction Manager).- Es un subsistema encargado de determinar que nodo deberá realizar cada operación a lo largo de una transacción.

4.10 OPTIMIZACION DE LA INTRANET

En el caso de que nuestro sistema distribuido (base de datos y/o aplicaciones) funcione total o parcialmente sobre una LAN o redes internas (INTRANET) que podamos administrar y configurar, podemos optar por la optimización de sistemas distribuidos propietario (interno) de cómputo y de

comunicación. Para ello se puede utilizar teoría de grafos y aplicar dos técnicas de optimización: las continuas (*métodos de gradiente, relajación lagrangeana y de penalización y programación lineal*) y las discretas (*exactos como Branch&Bound, planos de corte*). Para la selección de algoritmos se podrá utilizar la teoría de la complejidad que aplica dos métodos: los exactos y los aproximados (*algoritmos genéticos, simulatead annealing, ant systems y tabu search*).

Para aplicar cualquiera de estos u otros métodos se requiere familiaridad con problemas de: asignación, rutas más cortas y agente viajero, transporte y transbordo, flujo máximo y flujo mínimo. Además, deben ser analizados los problemas de *optimización combinatoria* como asignación de tareas a procesadores, distribución óptima de fragmentos en sistemas multibase de datos, distribución óptima de capacidades, etc. Mientras mas elementos se afinen dentro de nuestro proceso de optimización mejor será el resultado final, el consecuencia no debe pasarse por alto ningún elemento por mas insignificante que parezca en la incidencia del rendimiento.

CAPITULO V

PROPUESTA PARA LA IMPLEMENTACIÓN DE DIRECTRICES

5.1 INTRODUCCION

Para probar la hipótesis en esta investigación se desarrolló un caso práctico el cual contempla la elaboración de un sistema de comercio electrónico que consta de una base de datos distribuida y una aplicación de comercio electrónico, sobre el cual se analiza el impacto de las directrices propuestas. Vale indicar que no se presentan con detalle las etapas de desarrollo que generaría un proceso metodológico de ingeniería de software a estos componentes (BDD y aplicación), por no ser el objetivo primordial de esta investigación, sino mas bien, a los resultados que daría dichos procesos se aplicarán los procesos de optimización propuestas en esta investigación. Es a ésta última parte, a la que dedicaremos principal atención por la importancia que implica el análisis de las directrices, su impacto y repercusión en la eficiencia de sistemas de comercialización virtual.

5.2 OBJETIVOS

- Diseñar un prototipo de un sistema de comercio electrónico, el cual sirva de escenario para la aplicación de directrices de optimización.
- Aplicar las directrices de optimización sobre los componentes del prototipo del sistema de comercio electrónico, esto es: la base de datos distribuida y la aplicación de comercio electrónico.

- Realizar configuraciones de de cada uno de los componentes hardware y software de acuerdo a las directrices planteadas y a las características del entorno.
- Evaluar los resultados de la aplicación de las directrices de optimización.
- Redactar un conjunto de directrices ejecutivas cuya aplicación sea práctica.

5.3 REQUERIMIENTOS PARA LA IMPLEMENTACIÓN

Una red esta compuesta de varios componentes hardware, software y de comunicación, en este apartado especifican únicamente los componentes relevantes para nuestro propósito. Para esto se utilizó las instalaciones físicas de COMPUNET, cuyas características se describen a continuación:

5.3.1 SOFTWARE.

- Sistemas operativos para los servidores de base de datos y aplicaciones.- Microsoft Windows NT 4.0
- Sistemas operativos para los clientes.- Microsoft Windows 98se, Me, Xp
- DBMS distribuido.- Microsoft SQLServer 2000 con sus extensiones de servidor y cliente.
- Interfaz de Bases de Datos.- Controladores ODBC 3.70.08.20 para SQLServer.
- Herramienta de desarrollo de la Aplicación Web.- Microsoft Visual Interdev 6.0.
- Aplicación Web.- Aplicación desarrollada en Visual Interdev 6.0 que sirve de front end del sistema de comercio electrónico.
- Software de conexión.- Protocolo TCP/IP para la configuración de la red.

5.3.2 HARDWARE

- Terminales 1: 2 equipos Intel Celeron de 700 MHZ, 20 GB, 128 MB, 10/100.
- Terminales 2: 3 equipos Intel Celeron de 800 MHZ, 40 GB, 256 MB, 10/100.
- Servidores: 2 equipos Intel Pentium 4 de 1.8 GHZ, 80 GB, 512 MB, 10/100.

- Concentrador: Switch 3Com, 16Puertos RJ-45, 10/100.
- Hardware de Conexión: Cableado estructurado categoría 5

5.4 METODO PARA LA REALIZACION

Para la realización de las pruebas se va a utilizar un método experimental secuencial por etapas con retroalimentación de información en cada una de ellas. Para ello se ha organizado las etapas de la siguiente manera:

1. Escenario del sistema de comercio electrónico
 - a. La base de datos
 - b. La aplicación de comercio electrónico
 - c. Configuración del Servidor Web y de Base de Datos
 - d. Configuración de los orígenes de datos ODBC
2. Población y Muestra
3. Métodos e instrumentos de recolección de datos
4. Técnicas de procesamiento y análisis de datos
 - a. Definición de métricas
 - b. Evaluación de métricas
5. Procesamiento y análisis de estadísticas de rendimiento
6. Prueba de la hipótesis
7. Descripción de directrices ejecutivas

5.4.1 ESCENARIO DEL SISTEMA DE COMERCIO ELECTRONICO

El sistema de comercio electrónico permite que los usuarios puedan navegar por la tienda virtual, seleccionar los productos, modificar y confirmar su pedido, y cancelar su compra. Por el lado del proveedor se encarga de verificar los pedidos, los pagos y despachar los productos. Esta es una rápida

descripción de lo que hace un sistema de comercio virtual. Para representar este escenario, se desarrollo un prototipo, por lo que se debe considerar como tal.

5.4.1.1 LA BASE DE DATOS

Como herramienta de diseño se utiliza la Erwin 3.5.2 con Service Pack 2 de Platinum Technology.

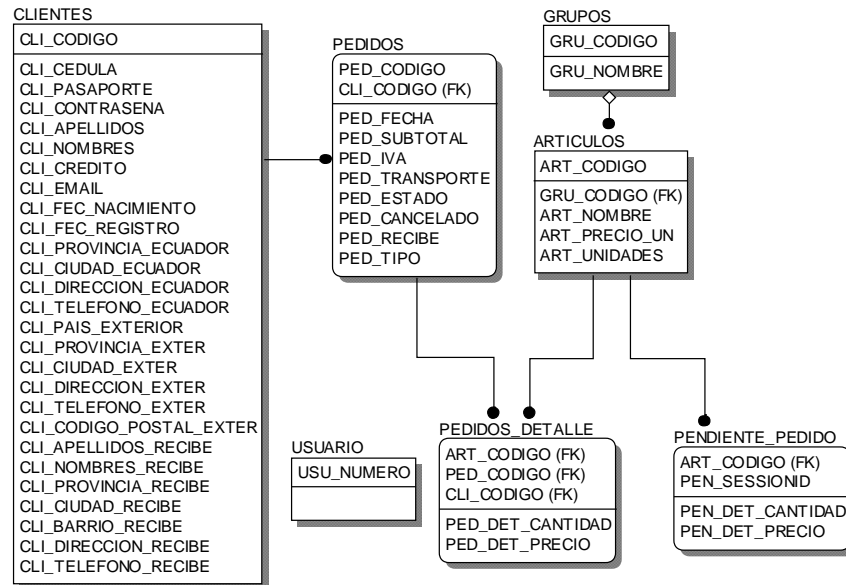


Fig. 20.- Diseño de la base de datos

Este diseño se implementó en Microsoft SQL Server 2000, sobre servidores Windows NT 4.0 con Service Pack 4.0, utilizando TCP/IP como protocolo de comunicaciones. Las características de configuración de estos componentes se presentan más adelante.

5.4.1.2 LA APLICACIÓN DE COMERCIO ELECTRONICO

Como plataforma de desarrollo de la aplicación se utiliza la Herramienta Microsoft Visual Interdev 6.0. El código Fuente de la aplicación esta en el Anexo III. El diagrama de flujo que se utiliza el prototipo se muestra a continuación:

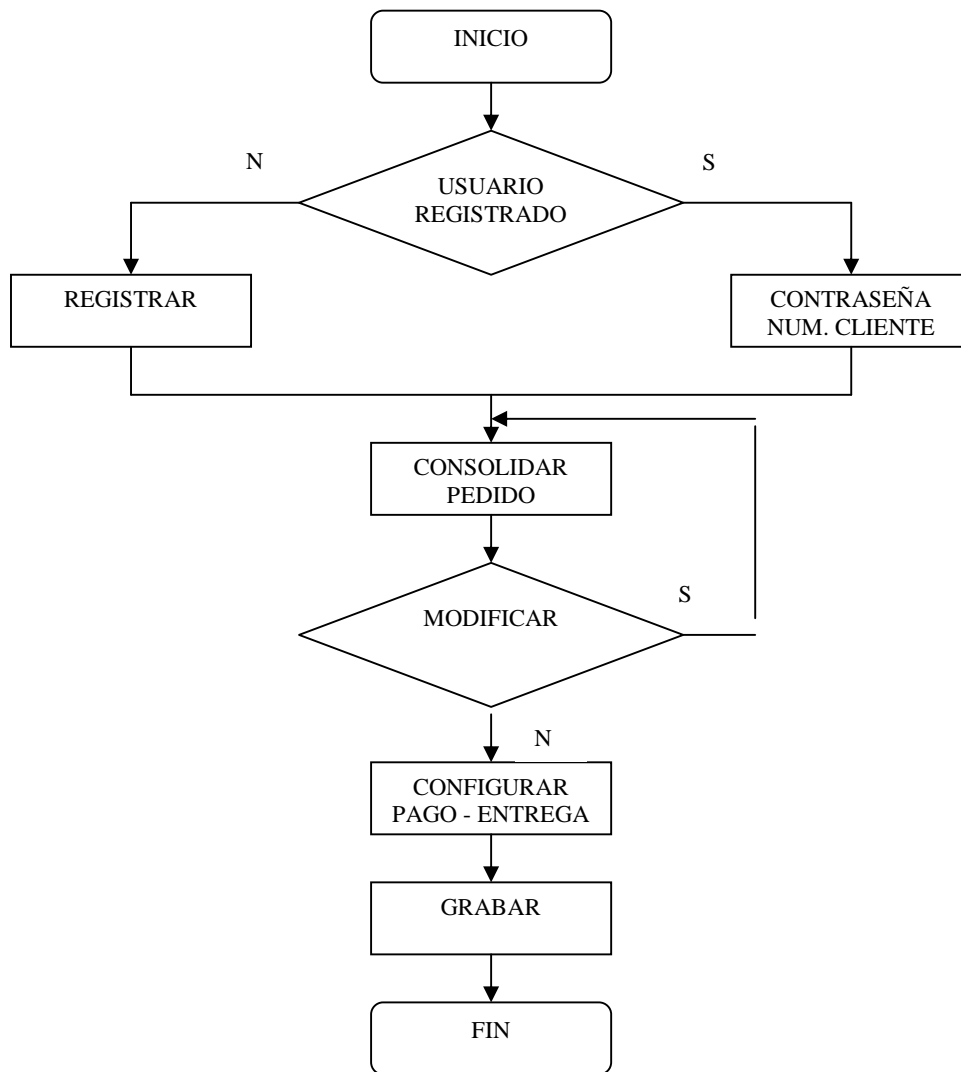


Fig. 21.- Diagrama del prototipo de la aplicación de comercio electrónico

Partiendo de la estructura anterior vamos a estudiar cada una de las etapas para la realización de la compra. El conjunto de pantallas que conforman la interfaz principal y necesaria para cumplir una negociación virtual se muestra a continuación:

1. *Página de Inicio (Home Page)*.- En la parte que nos incumbe, esta solicita información a cerca del usuario como es su número de registro y su contraseña, permitiendo de esta manera dar un acceso

inmediato al catálogo de los productos. Además presenta información introductoria a cerca de la empresa, su misión, visión y otros datos informativos.



Fig. 22.- Prototipo página de inicio

2. *Página de Registro.*- Previo a ingresar al catálogo de productos e ingresar al sistema on-line, el cliente debe registrarse. Este paso es necesario ejecutarlo una sola vez por cada cliente. Para ello se utiliza una interfaz como la siguiente:

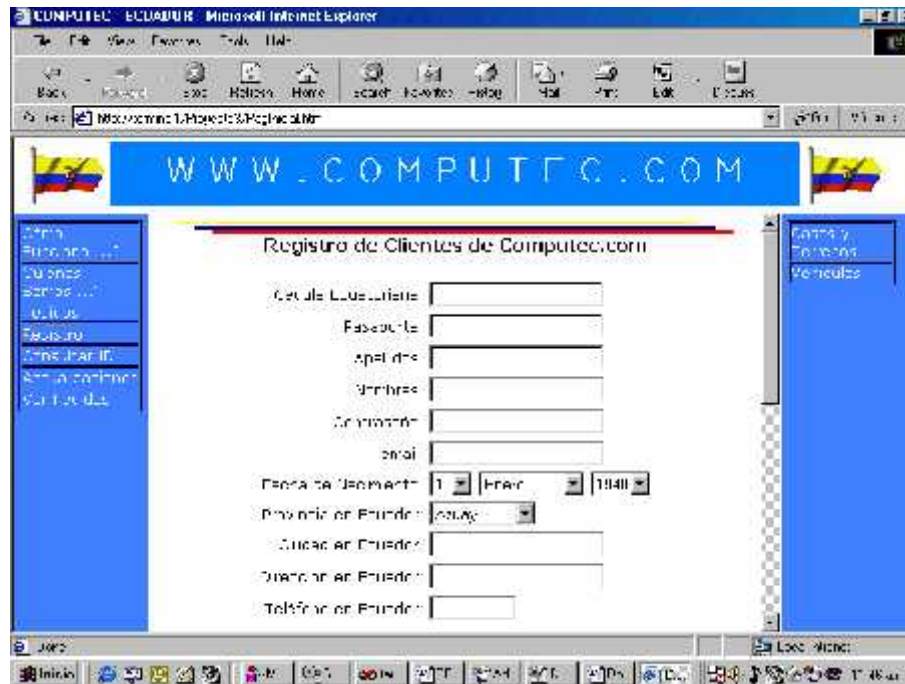


Fig. 23.- Prototipo página de registro

3. *Página de pedidos (Shopping Car).*- Permite elegir los productos que se desea adquirir, para ello se categorizar los mismos de acuerdo a su tipo, y se da a elegir de una lista desplegable, lo que mejora la interoperabilidad, ya que ello permite realizar una transferencia de parámetros por entrada estándar, lo que su vez minimiza el volumen de información de retorno por la selectividad minitérmino.

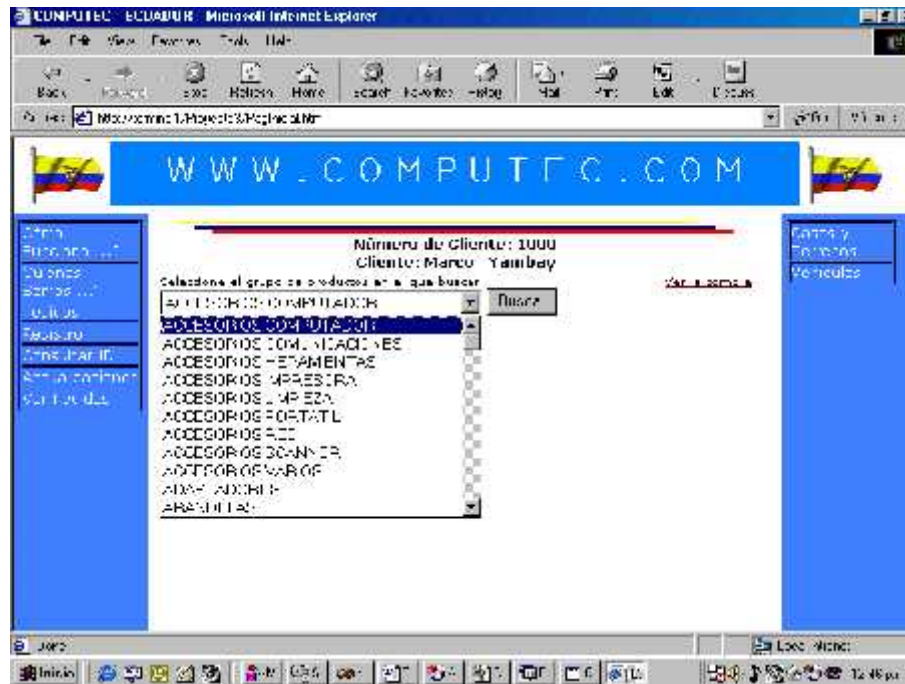


Fig. 24.- Prototipo página de pedidos

Esto disminuye el rango de búsqueda para la siguiente lista, ya que ésta desplegará solo los productos de esa categoría. Al igual que él, la selección de la categoría, esto permite mejor tiempo de respuesta ya que la transferencia de parámetros es por entrada estándar.

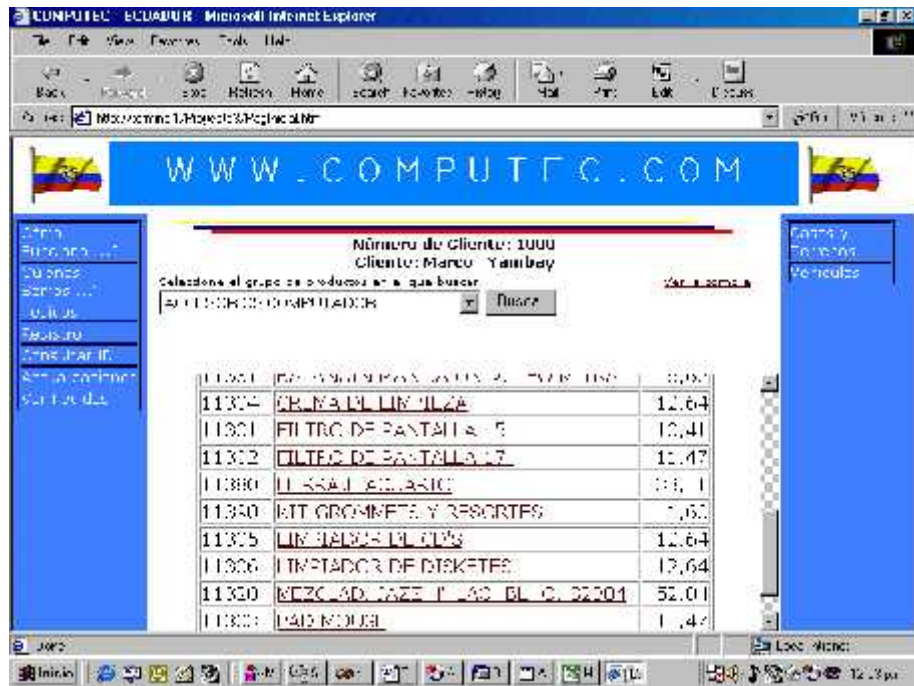


Fig. 25.- Prototipo segmentación del pedido

Para el producto seleccionado se solicita la cantidad que desea adquirir.



Fig. 26.- Prototipo página de pedido cantidad

4. *Página de Consolidación de Pedido.*- Una vez que el proceso de selección de productos y sus cantidades es definido se presenta un previo del pedido con los productos que se hayan seleccionado en la “Página de Pedidos”, a partir de la cual el usuario podrá modificar el mismo, eliminando productos, modificando la cantidad o producto, o, añadiendo más productos.

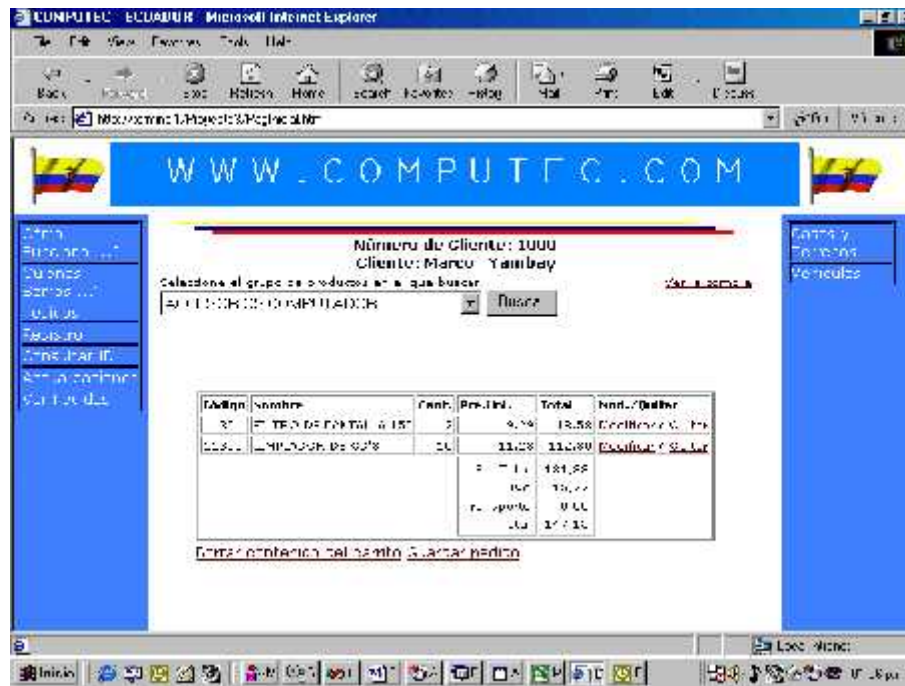


Fig. 27.- Prototipo página consolidación del pedido

5. *Página de Forma de Pago.*- Una vez que el pedido está a satisfacción del cliente, cuando el cliente desea adquirir viene la parte de pago, donde el usuario define el mecanismo para cancelar su compra.



Fig. 28.- Prototipo página forma de pago

Para nuestro estudio, este prototipo, ejemplifica las etapas principales dentro de un sistema de comercio electrónico. Por el lado del proveedor se puede diseñar de igual manera una aplicación Web de consultas para que indique el estado en que se encuentra cada una de las solicitudes de compra.

5.4.1.3 CONFIGURACION DEL SERVIDOR WEB Y DE BASE DE DATOS

Para la Configuración del servidor Web, es necesario habilitar el servicio WWW de Windows NT, el cual se instala como servicios Internet. Para ello se debe especificar, el Controlador de SQL Server.

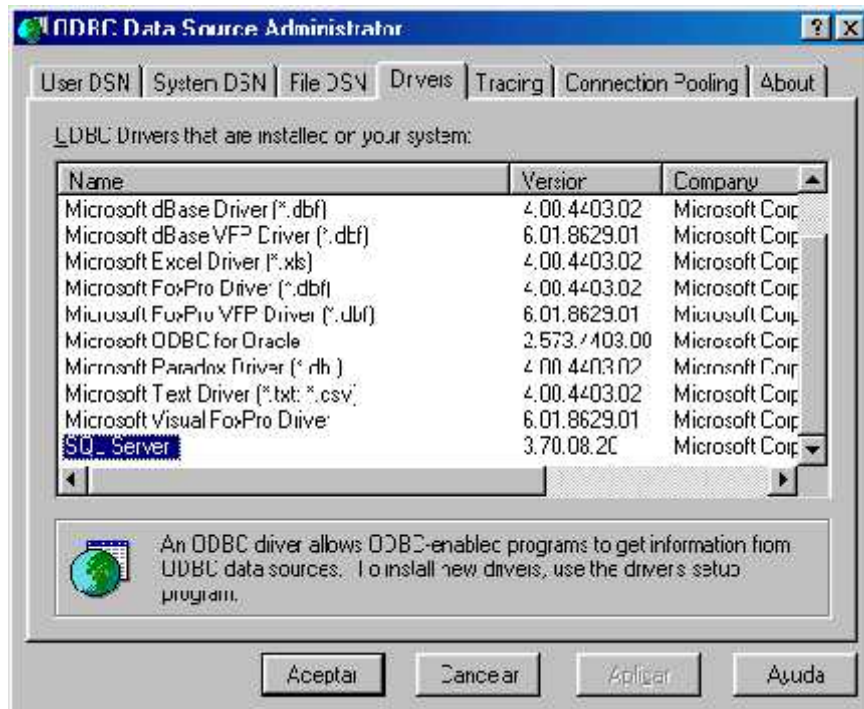


Fig. 29.- Configuración del servidor WEB

Posteriormente se procede a crear la fuente de datos para SQLServer, para ello hay que definir el tipo de autenticación que se va a usar.

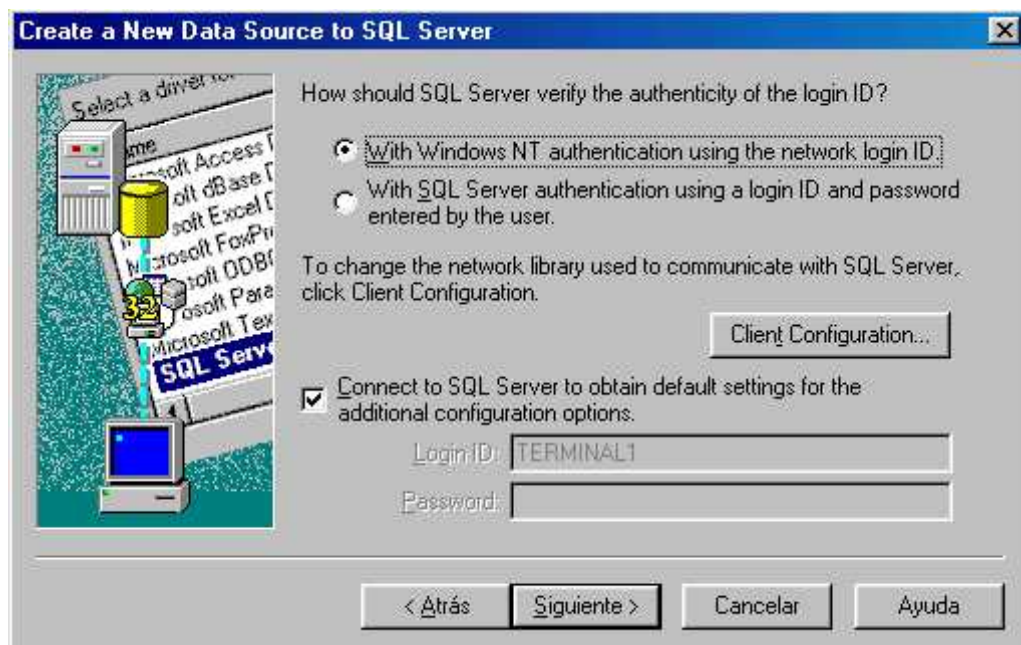


Fig. 30.- Autenticación para SQL SERVER

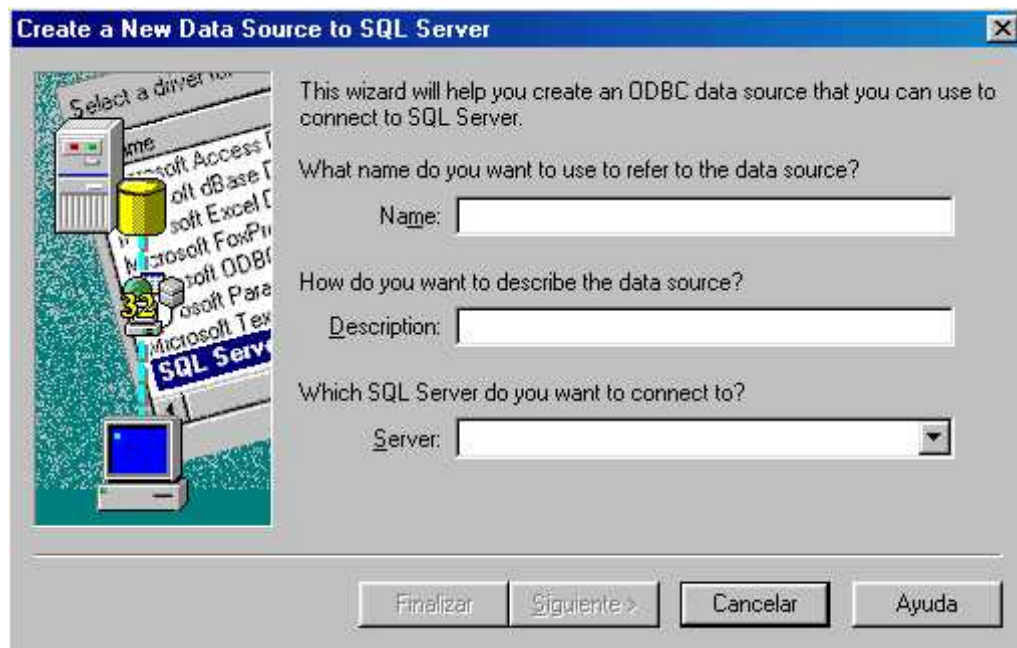


Fig. 31.- Creación de fuente de datos

Una vez instalado el ODBC se procede a configurar Servicio Web:

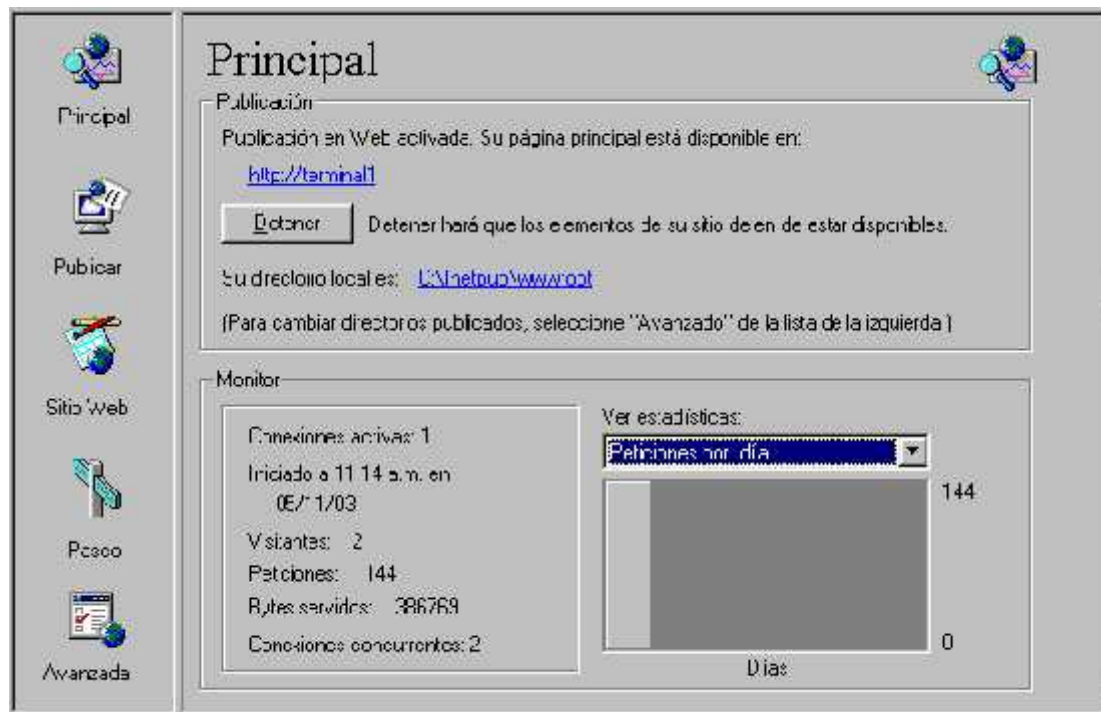


Fig. 32.- Configuración del servicio WEB

Por el lado del cliente, se necesita un navegador, como el Explorer o Netscape

5.4.1.4 CONFIGURACION DE LOS ORIGENES DE DATOS ODBC

Cuando se selecciona la opción ODBC durante la instalación, se instalan los componentes de ODBC para diferentes bases de datos. Desde la versión 2.5 de ODBC acepta *Nombres de Origen de Datos (DSN.- Data Source Name)* del sistema que es necesaria para conectar al servicio del ODBC de un sistema operativo para una base de datos en particular. Ej. El DNS en Windows NT permite utilizar ODBC a través del servicio de Microsoft Internet Information Server (IIS Services).

El DSN es un nombre lógico utilizado por el ODBC para hacer referencia al controlador y a cualquier otra información necesaria para tener acceso a los datos. Esta información indica el nombre real del servidor, la ubicación de la base de datos entre otras. En los archivos del conector de bases de datos de Internet (.idc), el Nombre del Origen de Datos (DSN) se utiliza para informar al Servicio ODBC acerca de donde se accederán a los datos de un proveedor en particular. Por ejemplo para Microsoft SQL Server se indicará, el nombre del servidor, la dirección de red y la biblioteca de red específicos de su instalación.

5.4.2 POBLACION Y MUESTRA

A partir del escenario descrito anteriormente se define:

- *La población.-* Para nuestro análisis ésta representa un sistema de comercio electrónico que incluye todos los requerimientos de funcionalidad descritos en un SRS comercial.
- *La muestra.-* Para ello se diseñó un prototipo de sistema de comercio electrónico que incluye ciertas características funcionales para la realización de nuestras pruebas.
- *El porcentaje que representa la muestra.-* Basado en la experiencia en el desarrollo y utilización de sistemas de comercio electrónico, y tomando en cuenta frecuencia de uso de las funciones de sistemas de esta naturaleza se considera que éste contempla un 42% de funcionalidad.

5.4.3 METODOS E INSTRUMENTOS DE RECOLECCION DE DATOS

Para la recolección de datos se utilizó un método experimental, el cual contempla varias evaluaciones de métricas de rendimiento durante la ejecución de un mismo sistema de comercio electrónico, pero con la base de datos en diferentes estados de distribución. Adicionalmente se reforzó con consulta bibliográfica en libros, manuales, folletos, revistas, estudios previos y otros medios que presenten información referente a nuestra investigación, consultas en Internet y consultas a especialistas.

Para realizar estas pruebas se utiliza como instrumentos de recolección de datos un prototipo de sistema de comercio electrónico compuesto de una aplicación Web para comercio electrónico y una base de datos distribuida.

La aplicación Web embebe un temporizador, el cual evalúa las métricas definidas los índices de rendimiento del sistema. La base de datos por su parte presenta su esquema en diferentes etapas del proceso de distribución.

5.4.4 TECNICAS DE PROCESAMIENTO Y ANALISIS DE DATOS

Como técnica de procesamiento de datos se realizó una tabulación de la información obtenida mediante los métodos e instrumentos de recolección de datos, las mismas que sirvieron para realizar su posterior análisis. Para el análisis de la información recolectada se utilizó un estudio estadístico de la información, obteniendo las interpretaciones de cada uno de los análisis. Los resultados de estos procedimientos se mencionan a continuación.

5.4.4.1 DEFINICION DE METRICAS

Como se menciona en el capítulo cuarto de esta investigación, las métricas que se deben utilizar para la evaluación de rendimiento del sistema de comercio electrónico son:

1. El tiempo de respuesta, que representa el intervalo de tiempo entre el envío de la solicitud por parte de la aplicación cliente y el retorno de su respuesta emitida por el servidor. Como se puede ver, ésta es una métrica que evalúa la eficiencia del sistema desde la perspectiva de la aplicación cliente.
2. Throughput: número de trabajos procesados por el servidor en un intervalo de tiempo. Como se puede ver, ésta es una métrica que evalúa la eficiencia del sistema desde la perspectiva del servidor en conjunto como servidor de aplicación, servidor http, y servidor de base de datos.

Cabe indicar que la segunda métrica (throughput) esta embebida en el tiempo de respuesta, por lo que al evaluar a esta última se obtiene una evaluación indirecta de throughput. En base a estas dos métricas se analizó el comportamiento del sistema antes, durante y después de aplicar el proceso de optimización descrito en el cuarto capítulo de ésta investigación.

5.4.4.2 EVALUACION DE METRICAS

Para garantizar la certeza de las medidas, las pruebas se realizaron sobre diferentes componentes del sistema de comercio electrónico:

a. Base de datos.- Base de datos en diferentes etapas del proceso de distribución:

BD1.- Base de datos sin fragmentación (centralizada)

BD2.- Base de datos fragmentada por entidades

BD3.- Base de datos fragmentada verticalmente

BD4.- Base de datos fragmentada horizontalmente

BD5.- Base de datos con fragmentación mixta

b. Elementos funcionales.- Transacciones con diferentes niveles de requerimientos de información:

1. Registro, actualización y consulta de clientes

2. Registro, actualización y consulta de pedidos

c. Host.- Las pruebas se implementaron desde dos tipos de host.

1. Servidor (host local).- Cuando el equipo que funge de servidor ejecuta aplicaciones de cliente

2. Cliente (host remoto).- Cuando el equipo que funge de cliente ejecuta aplicaciones de cliente

Como se indica en el numeral anterior, la métrica de tiempo de respuesta embebe al throughput. Los valores presentados en las siguientes tablas representan el tiempo promedio de ejecución de cada transacción en un determinado host y sobre una base de datos determinada.

PEDIDOS			
SERVIDOR			
TIEMPO DE EJECUCION DE TRANSACCIONES (ms)			
	INGRESO	CONSULTA	ACTUALIZACION
BD1	61	11	52
BD2	56	9	49
DIFERENCIA	5	2	3

CLIENTE			
TIEMPO DE EJECUCION DE TRANSACCIONES (ms)			
	INGRESO	CONSULTA	ACTUALIZACION
BD1	65	13	44
BD2	58	10	42
DIFERENCIA	7	3	2

CLIENTES

SERVIDOR

TIEMPO DE EJECUCION DE TRANSACCIONES (ms)

	INGRESO	CONSULTA	ACTUALIZACION
BD1	48	8	22
BD3	45	7	14
DIFERENCIA	3	1	8

CLIENTE

TIEMPO DE EJECUCION DE TRANSACCIONES (ms)

	INGRESO	CONSULTA	ACTUALIZACION
BD1	41	7	24
BD3	38	6	15
DIFERENCIA	3	1	9

CLIENTES

SERVIDOR

TIEMPO DE EJECUCION DE TRANSACCIONES (ms)

	INGRESO	CONSULTA	ACTUALIZACION
BD1	48	8	22
BD4	46	7	13
DIFERENCIA	2	1	9

CLIENTE

TIEMPO DE EJECUCION DE TRANSACCIONES (ms)

	INGRESO	CONSULTA	ACTUALIZACION
BD1	41	7	24
BD4	39	6	18
DIFERENCIA	2	1	6

PEDIDOS

SERVIDOR			
TIEMPO DE EJECUCION DE TRANSACCIONES (ms)			
	INGRESO	CONSULTA	ACTUALIZACION
BD1	61	11	52
BD5	69	15	57
DIFERENCIA	-8	-4	-5

CLIENTE			
TIEMPO DE EJECUCION DE TRANSACCIONES (ms)			
	INGRESO	CONSULTA	ACTUALIZACION
BD1	65	13	44
BD5	76	16	46
DIFERENCIA	-11	-3	-2

Tab. 11.- Tiempo de ejecución de transacciones

5.5 PROCESAMIENTO Y ANALISIS DE LAS ESTADISTICAS DE RENDIMIENTO

A partir de las tabulaciones anteriores, para una mejor interpretación de los resultados se procede a representarlos gráficamente mediante cuadros estadísticos. Las estadísticas están divididas de acuerdo al host donde se ejecutó las pruebas las mismas que fueron ejecutadas sobre los procesamientos más representativos respecto a las transacciones.

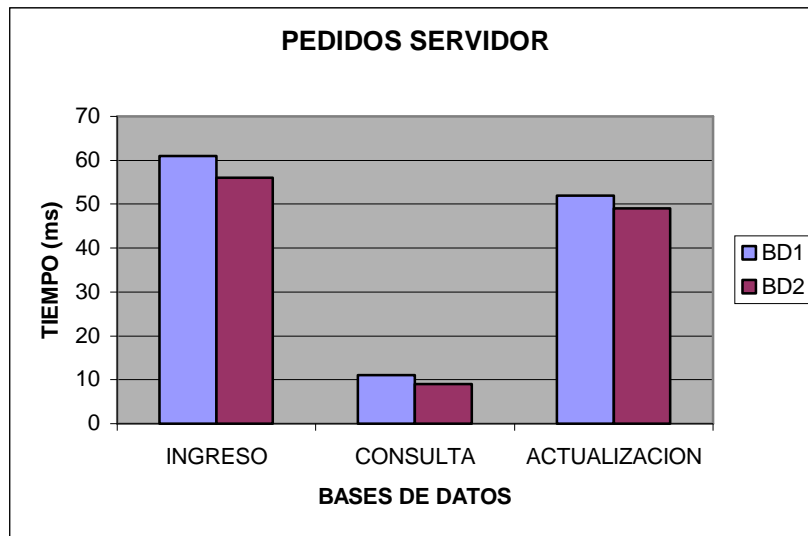


Fig. 33.- Transacciones de pedidos en servidor

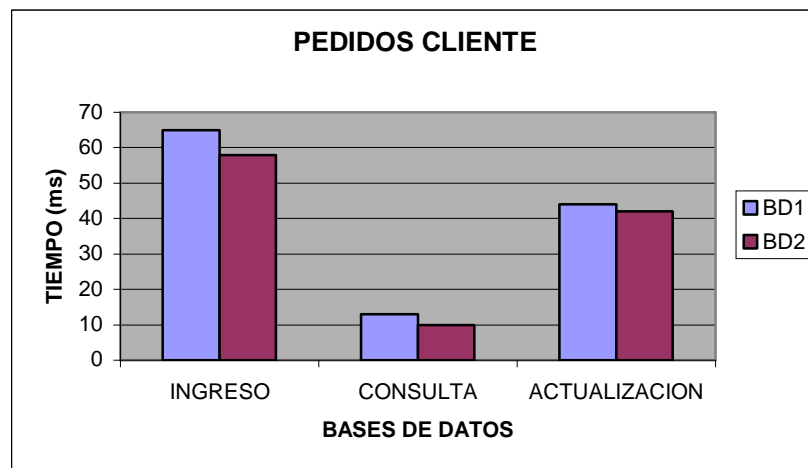


Fig. 34.- Transacciones de pedidos en cliente

INTERPRETACION 1.- De los dos cuadros anteriores, se puede ver que este tipo de fraccionamiento (por entidades) no es el adecuado cuando se requiere un rendimiento balanceado del sistema, ya que si bien favorece a las consultas (en comparación con las estadísticas posteriores), el ingreso y la actualización implican mayor tiempo (en comparación con las estadísticas posteriores).

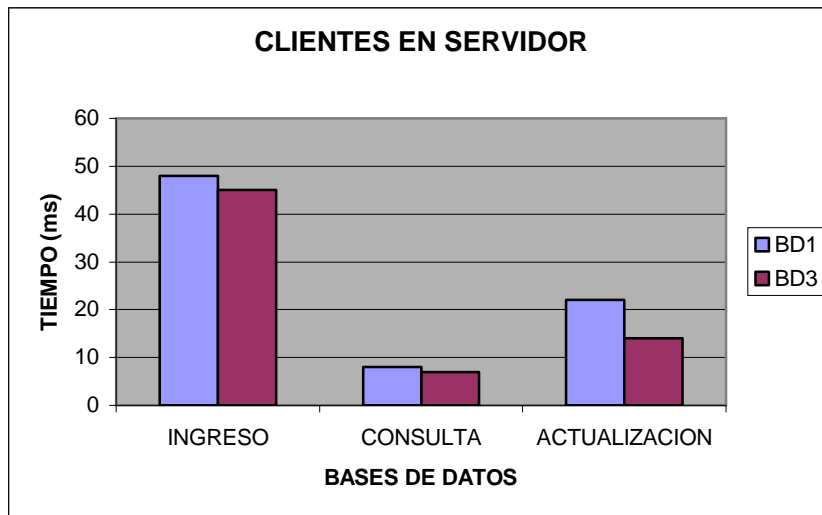


Fig. 35.- Transacciones clientes en servidor

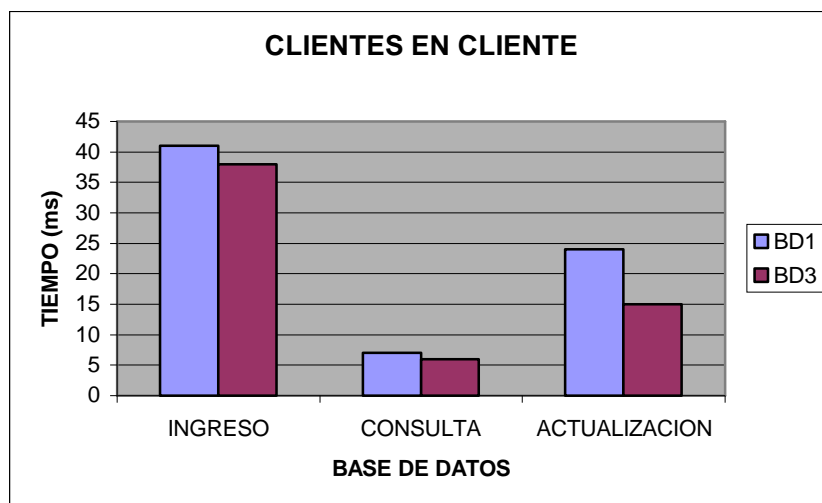


Fig. 36.- Transacciones clientes en clientes

INTERPRETACION 2.- De los dos cuadros anteriores, se puede ver que este tipo de fraccionamiento (vertical) responde de mejor manera en cuanto a un rendimiento homogéneo del sistema, ya que favorece a todas las transacciones.

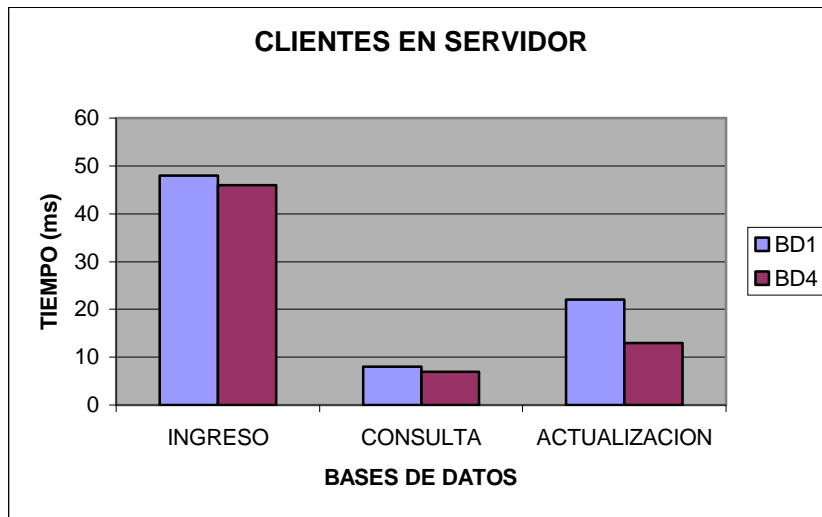


Fig. 37.- Transacciones de clientes en servidor fraccionamiento vertical

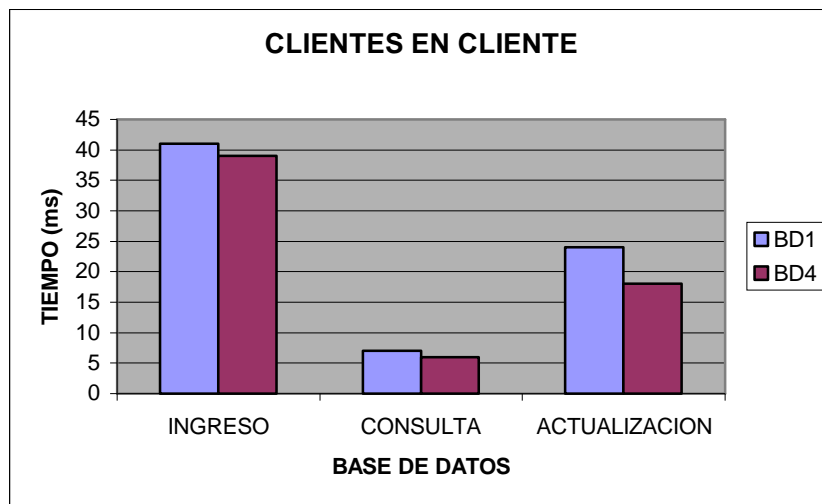


Fig. 38.- Transacciones de clientes en cliente fraccionamiento vertical

INTERPRETACION 3.- De los dos cuadros anteriores, al igual que en la segunda interpretación, se puede ver que este tipo de fraccionamiento (horizontal) tiene un rendimiento homogéneo del sistema, favoreciendo a todas las transacciones.

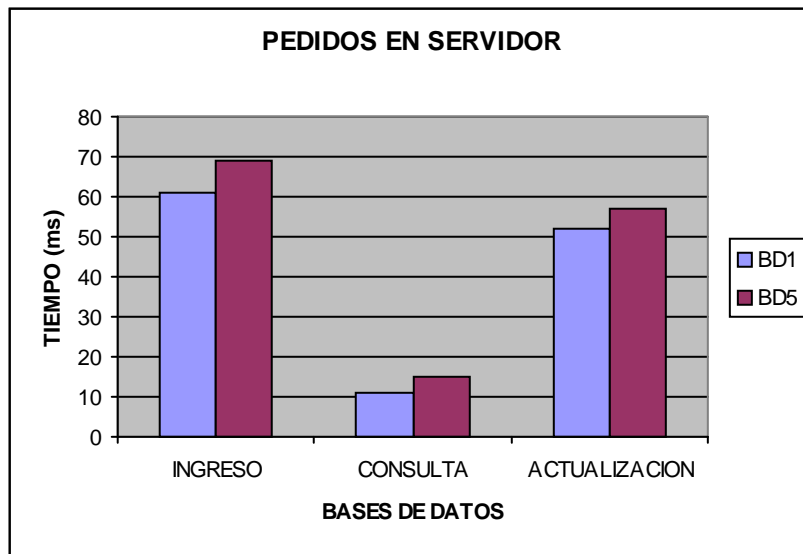


Fig. 39.- Transacciones de clientes en servidor fraccionamiento horizontal

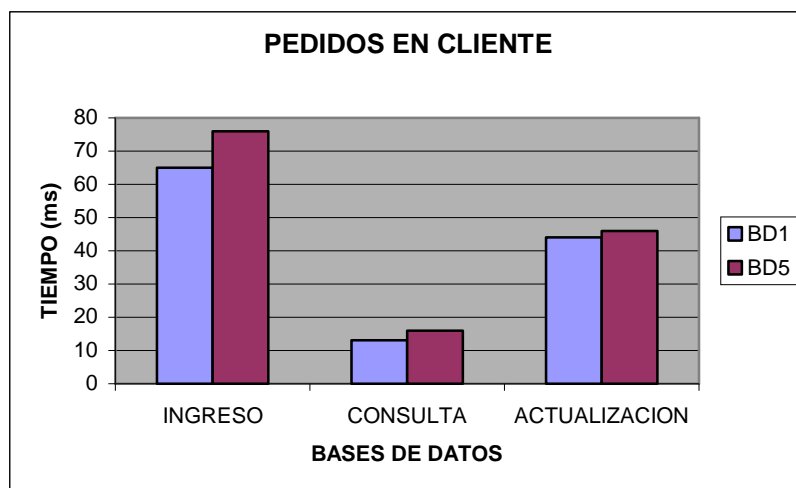


Fig. 40.- Transacciones de clientes en cliente fraccionamiento horizontal

INTERPRETACION 4.- De los dos cuadros anteriores, se puede analizar que para este caso, este tipo de fraccionamiento (híbrida) incrementa los tiempos de respuesta de todas las transacciones penalizando el rendimiento del sistema.

5.6 PRUEBA DE LA HIPOTESIS

Los argumentos teóricos expuestos en esta investigación determinan los mecanismos técnicos adecuados para realizar un proceso de distribución de forma eficiente. Los mecanismos contemplan los factores relevantes en el entorno del sistema de comercio electrónico. Los factores son procesados mediante procedimientos técnicos para establecer el mejor índice de rendimiento global.

Es la aplicación de este conjunto de mecanismos los que garantizan que los resultados apunten a la optimización del sistema de comercio electrónico.

Con la ayuda de un prototipo de sistema de comercio electrónico, en el cual se analiza las estadísticas de rendimiento, y para sustentar la afirmación de la hipótesis planteada, se concluye que:

1. Al aplicar adecuados procedimientos de distribución a la base de datos centralizada (BD1) se obtiene un incremento en el rendimiento global del sistema de comercio electrónico.
2. Cuando el nivel de distribución no es el apropiado (BD3, BD4), las penalidades en una base de datos distribuida mal diseñada (BD5) son mayores que los beneficios.
3. La degradación o incremento del rendimiento del sistema de comercio electrónico depende del grado de interoperabilidad de la base de datos distribuida, alcanzando su máxima eficiencia en la ejecución de las transacciones y obteniendo los mejores tiempos de respuesta cuando su diseño es el adecuado al entorno en el que se desenvuelve el sistema.
4. En consecuencia, el sistema de comercio electrónico al utilizar una base de datos cuya operación interna es eficiente, optimiza su rendimiento. Dicha optimización depende de las componentes funcionales con que cuente cada sistema en particular.

CAPITULO VI

PROPUESTA DE DIRECTRICES EJECUTIVAS

Cabe recalcar que la aplicación de directrices de interoperabilidad a una base de datos distribuida, en aras de optimizar el rendimiento de un sistema de comercio electrónico, es un proceso continuo durante su diseño y desarrollo. Los numerales siguientes indican las directrices propuestas en esta investigación, las cuales deben ser consideradas como línea base para futuras procesos de depuración y mejoramiento continuo.

6.1 DIRECTRICES DE CARATER GENERAL

1. Se necesita de una monitorización continua y de ajustes periódicos, para que si se llegan a detectar o producir desviaciones, se pueda aplicar procesos de re-ingeniería e ingeniería inversa para corregir dichos inconvenientes.
2. El proceso de distribución y la aplicación de directrices de optimización se justifica cuando un sistema de comercio electrónico no alcanza un rendimiento homogéneo y aceptable.
3. Los sistemas de comercio electrónico son aplicaciones que utilizan procesos transaccionales complejos, en el que intervienen varias instancias, por lo que para cada proyecto se debe llevar a cabo un análisis autónomo de acuerdo a sus características, considerando las necesidades de información estudiadas en el capítulo cuarto de esta investigación.

6.2 DIRECTRICES DE CARÁCTER ESPECIFICO

1. Las vistas de las aplicaciones normalmente son subconjuntos de relaciones. Además, la localidad de sus accesos no está definida sobre relaciones enteras pero sí sobre subconjuntos de las mismas. Por ello en primera instancia, se debe considerar como unidad de distribución a subconjuntos horizontales o verticales de relaciones.
2. Si las aplicaciones tienen vistas definidas sobre una determinada relación (considerándola como una unidad de distribución) que reside en varios sitios de la red, se debe analizar que esta estrategia no cause problemas en la ejecución de las actualizaciones y el costo del espacio de almacenamiento.
3. Los beneficios de la replicación son inversamente proporcionales a la relación entre las operaciones de DML (ingreso, eliminación y actualización) y las operaciones SQL (consultas). En tal virtud, como regla, la replicación de fragmentos es de utilidad cuando el número de consultas de solo lectura es mayor que el número de consultas de actualizaciones.
4. Uno de los objetivos de la descomposición de una relación en fragmentos es permitir el proceso concurrente de las transacciones. Para que este beneficio sea maximizado, se recomienda la ejecución paralela de sentencias.
5. Para obtener el máximo rendimiento de la distribución de una base de datos, la aplicación de comercio electrónico debe incluir mecanismos que permita la ejecución paralela de transacciones al dividir las sentencias en una serie de sub-sentencias que operen sobre distintos fragmentos.
6. Si las aplicaciones tienen requisitos que limiten la descomposición de la relación en fragmentos mutuamente exclusivos, las vistas de estas aplicaciones estarán definidas sobre más de un fragmento, lo que puede causar una degradación en el rendimiento, debido a que será necesario recuperar los datos de dos fragmentos y llevar a cabo sobre ellos operaciones de unión y yunto.
7. Para el algoritmo de fragmentación horizontal se debe asegurar que el conjunto de predicados simples sea completo y mínimo.

8. El establecimiento de los predicados minitérmino es trivial, pero la dificultad radica en el tamaño del conjunto de predicados minitérmino, que es exponencial respecto al número de predicados simples. Para minimizarlos, es necesario reducir su número mediante el uso de implicaciones lógicas y mediante la aplicación de criterios de dependencia de transacciones.
9. La fragmentación horizontal derivada se define sobre una relación miembro de acuerdo a una operación de selección especificada sobre su propietaria. Para que el rendimiento sea homogéneo en las demandas a los datos se debe considerar que el enlace entre las relaciones propietaria y miembro se debe definir como un equi-yunto y que a su vez el equi-yunto puede desarrollarse a través de semiyuntos. Este segundo punto es especialmente importante para nuestros propósitos, ya que deseamos fraccionar una relación miembro según la fragmentación de su propietaria, además es necesario que el fragmento resultante se defina únicamente sobre los atributos de la relación miembro.
10. En un esquema de base de datos, resulta frecuente que exista más de dos enlaces sobre una relación R. En este caso, aparece más de una posibilidad de fragmentación horizontal derivada. La decisión para elegir una u otra se debe basar en dos criterios:
 - a. la fragmentación con mejores características de yunto y
 - b. la fragmentación empleada en más aplicaciones.
11. Se recomienda que los diseños donde la relación de yunto entre los fragmentos es simple, la asignación de la propietaria como de la miembro se ubiquen en el mismo sitio.
12. Se debe facilitar el acceso a los usuarios que hagan mayor uso de los datos para que de esta manera el impacto total del rendimiento del sistema sea mínimo.
13. La compleción de una fragmentación horizontal primaria se basa en la selección de los predicados a usar. En tal virtud, en la medida que los predicados seleccionados sean completos, el resultado de la fragmentación también lo será.
14. Se debe establecer la disyunción de la fragmentación tanto para la relación primaria como para la derivada. En el primer caso, la disyunción se garantiza en la medida en que los predicados minitérmino que determinan la fragmentación sean mutuamente exclusivos. En la fragmentación

derivada, sin embargo, implica un semi-yunto que añade complejidad. En general, no se debe juntar una tupla de una relación miembro con dos o más tuplas de una relación propietaria cuando estas tuplas se encuentran en fragmentos diferentes a los de la propietaria.

15. La réplica de los atributos clave es una gran ventaja, a pesar de los problemas que pueda causar, ya que permite mantener la integridad semántica. Se debe tener en cuenta que cada dependencia es, de hecho, una restricción que influye sobre el valor de los atributos de las respectivas relaciones en todo momento. También muchas de estas dependencias implican a los atributos clave de una relación.
16. Se debe diseñar la base de datos tal que los atributos clave sean parte de un fragmento que está ubicado en un sitio, y los atributos relacionados sean parte de otro fragmento asignado de ser posible en el mismo sitio, de tal manera que cada petición de actualización que requiere la verificación de integridad no necesite de una comunicación con otros sitios.
17. La réplica de los atributos clave de cada fragmento reduce la problemática de integridad, pero no elimina toda su complejidad, ya que la comunicación puede ser necesaria para las restricciones de integridad así como para el control de concurrencia que implican a las claves primarias.
18. Una posible alternativa a la réplica de los atributos clave es el empleo de identificadores de tuplas, que son valores únicos asignados por el sistema a las tuplas de una relación. Mientras el sistema mantenga los identificadores, los fragmentos permanecerán íntegros. Aunque, ésta es una alternativa, es una medida que no se recomienda ya que aumenta un atributo que no es parte de la semántica de la instancia.
19. En el caso que se necesite alcanzar un nivel de fragmentación mixta, se debe tener en cuenta que el número de niveles arbóreos incrementa drásticamente el tiempo de reconstrucción de la relación, por lo que se recomienda que el número de niveles no debería superar el par VH o HV. Ya que por ejemplo, en el coste que supondría realizar la unión o el yunto de una relación con fragmentación nivel 7, evidentemente sería muy elevado y ese aumento de rendimiento que se persigue al aplicar estas técnicas, quizás, no se produzca.

20. Además de analizar los requerimientos de información, necesarios para los procedimientos de optimización planteados, se recomienda realizar extrapolaciones en cuanto a la variación (generalmente incremento) de dichos índices, ya que esto permitirá alargar el tiempo de vida del sistema.
21. En el caso de que los requerimientos de información relacionados a datos cuantitativos, cuyo análisis y proyección resulta compleja se recomienda que al menos se investiguen los más importantes. La importancia la podemos interpretar en la regla "80/20", la misma que indica que el 20% de las sentencias existentes acceden al 80% de los datos.
22. Para bases de datos de tamaño considerable se recomienda configurar los servidores Web en equipos independientes de los equipos servidores de bases de datos, con buenos canales de comunicación.
23. Como se menciona en el capítulo anterior, el rendimiento del sistema de comercio electrónico depende en gran medida de la calidad de diseño de la base de datos, sin embargo existen factores externos a él, propios del Internet que pueden afectar su rendimiento, en tal virtud conviene analizar también el escenario en que se encuentra cada sistema.
24. En el proceso de optimización de un sistema de comercio electrónico, no solo deben considerarse elementos externos a la base de datos y aplicación Web, sino también, se debe considerar otros componentes propietarios susceptibles de configuración tal como los sistemas operativos, software administrador de base de datos, software de comunicaciones, Intranet, entre otros.
25. No existe relación de proporcionalidad entre el nivel de fragmentación y el rendimiento de una base de datos distribuida, ya que en nuestro análisis de eficiencia se obtuvo mejor rendimiento con la fragmentación vertical u horizontal que con la fragmentación mixta.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- Debido a que los sistemas de comercio electrónico dependen de servicios de intermediarios, los mismos que no pueden ser administrados directamente, no se puede establecer un único procedimiento de optimización de la interoperabilidad de la base de datos distribuida y la aplicación de comercio electrónico.
- Dependiendo del nivel de control sobre los componentes de terceros pueden aplicarse los lineamientos de optimización, sin embargo, se debe siempre analizar los efectos colaterales que dicha configuración pueda ocasionar en detrimento de otras aplicaciones que corren sobre los mismos equipos. Por lo tanto las métricas deben ser analizadas de acuerdo al escenario en el que se desenvuelve el sistema, como una realidad completamente diferente.
- Al aplicar los lineamientos expuestos en esta investigación se obtendrá mejoría en los índices de rendimiento del sistema de comercio electrónico, sin embargo este será proporcional al impacto que tiene dicho factor sobre el rendimiento del sistema global. En tal virtud, es preciso analizar el impacto costo / beneficio que representa tomar una u otra media.
- Uno de los pilares que hay que apuntalar en sistemas distribuidos es la disponibilidad de la red, ya que una debilidad importante de los mismos es que además de las fallas que pueden ocurrir con el negocio virtual en si, se incrementa la probabilidad de error por cuestiones inherentes a una red que como se sabe “una red si no se cae, no es red”.
- Aunque una base de datos centralizada favorece a unas transacciones (consultas) pero penaliza otras (ingreso y actualizaciones), sin embargo, ésta puede dar mejores resultados

que una base de datos distribuida, cuyo procedimiento o nivel de distribución es el inadecuado.

- En la actualidad una de las aplicaciones que más explota, las bondades de una base de datos distribuida bien diseñada, son las aplicaciones de comercio electrónico, ya que obtiene un mejor rendimiento (por la distribución) y una mayor tolerancia a fallos (por la replicación) que un sistema centralizado no podría prestar.

RECOMENDACIONES

- Para potenciar la participación del consumidor en el Comercio Electrónico deben definirse medidas adecuadas de protección a nivel mundial, independientemente de la jurisdicción o legislación aplicable, ya que estas lo resguardan frente a prácticas comerciales que podrían lesionarlo, estableciendo paralelamente medidas de generación de confianza que complementadas con las normas actualmente existentes en Ecuador proporcionan el marco adecuado para el desarrollo del comercio en línea.
- La autorregulación ha probado ser una buena medida cuando existe adecuada competencia y diversidad de opciones, en este contexto es necesario establecer líneas base de rendimiento, para estar al tanto de los niveles que brindan las demás empresas, para someter nuestro sistema de comercio electrónico a procesos de calidad y mejoramiento continuo.
- Antes de la etapa de implantación, es necesario analizar por un lado los volúmenes de información tráfico, frecuencia, demanda, entre otras, para conocer el entorno en el cual se va a ejecutar el sistema. Por otro lado se debe analizar las ventajas de cada uno de los elementos de la plataforma software a utilizar para determinar las facilidades que cada una de ellas presta y poder hacer la mejor elección de acuerdo a las características del sistema de comercio electrónico.
- Es necesario agotar todas las alternativas de optimización software antes de pensar en repotenciones de hardware, ya que estas son mucho más costosas.

- Cuando los usuarios que utilizan el sistema de comercio electrónico se encuentran en ubicaciones geográficamente distantes (distintos continentes) o en lugares con condiciones diferentes en cuanto a prestaciones de Internet, un sistema de comercio electrónico con una base de datos centralizada no balancea los tiempos de respuesta, en estos casos es recomendable dotar al sistema en cuestión de una base de datos distribuida.

BIBLIOGRAFIA

BIBLIOGRAFÍA GENERAL:

- Diccionario de Computación - 2000 Términos, Freedman McGrawhill
- Leiva Zea F., Nociones de Metodología de Investigación Científica.
- Álvarez P (2000b), “Ocho de cada diez nuevos empleos se crearán en el sector tecnológico”, Cinco Días, 6/06/2000: 34-35.
- Andersen Consulting, e-Europe Takes off. Andersen Consulting, 1999.
- BusinessWeek, The new economy (especial report), january 31: 36-49 2000.
- Cachón L., Los nuevos yacimientos de empleo, Cuadernos de Información Económica, 151: 85-93, 1999.
- DELOITTE & TOUCHE, El comercio electrónico transforma la gestión del negocio, Banca & Finanzas, 45: 45, 1999.
- KPMG, El crecimiento del comercio electrónico, Banca & Finanzas,. 45: 43-44, 1999.
- Melle M., Informe de la OCDE sobre el impacto económico y social del comercio electrónico, Cuadernos de Información Económica, 144/145: 97-102, 1999.

BIBLIOGRAFÍA ESPECIALIZADA:

- Özsü M.T., Valdúriez P., Principles of Distributed Database 2da. Ed. Systems, Prentice Hall, 2001.
- Navathe S.B., Ceri S., Wiederhold G., Dou J., Vertical Partitioning of Algorithms for Database Design 3th. Ed., ACM, 1998.
- Jain A., Dubes R., Algorithms for Clustering Data, Prentice Hall Advanced Series, 1998.

- ❑ Navathe S.B., Karlapalem K., Minyoung Ra., A Mixed Fragmentation Methodology for Initial Distributed Database Design, 1997.
- ❑ Karlapalem K., Ng Moon Pun, Query-Driven Data Allocation Algorithms for Distributed Database Systems, DEXA 1997.
- ❑ Smith, J.K., Survey Paper on Vertical Partitioning, Universidad de Hawaii, 1997.
- ❑ Shepherd J.A., Harangsri B., Chen H.L., Ngu A.H.H., A Two-Phase Approach to Data Allocation in Distributed Databases, Fourth International Conference on Database Systems for Advanced Applications, World Scientific Press, Singapur, Singapur 1995, 1996.
- ❑ Brunstrom A., Leutenegger S.T., Simha R., Experimental Evaluation of Dynamic Data Allocation Strategies in a Distributed Database With Changing Workloads, Fourth International Conference on Information and Knowledge Management, noviembre 1995.
- ❑ Roger Pressman, Ingeniería del software, un enfoque practico, 1995, McGraw Hill.
- ❑ Navathe S.B.; Ceri S.; Wiederhold G.; Dou J. Vertical Partitioning of Algorithms for Database Design. ACM, 1994.
- ❑ Muthuraj J., Chakravarthy S., Varadarajan R., Navathe S.B., A Formal Approach to the Vertical Partitioning Problem in Distributed Database Design. PDIS, 1993.
- ❑ Bell D., Grimson J., Distributed Database Systems, Addison – Wesley, 1992.

DIRECCIONES INTERNET

- ❑ Web Services Interoperability Organization.- <http://www.ws-i.org/>
- ❑ Consultora Jupiter Communications, <http://www.ebusinessforum.com>
- ❑ International Data Corporation (IDC).- <http://www.commerce.net>
- ❑ El reto del comercio electrónico.- <http://www.marketingycomercio.com>
- ❑ IEEE Wireless Standards Zone.- <http://standards.ieee.org/wireless/>
- ❑ Hiperlan2 Global Forum.- <http://www.hiperlan2.com/>
- ❑ Atheros Communications.- <http://www.atheros.com/>
- ❑ The Economist Intelligence Unit (EIU) <http://www.ebusinessforum.com>

- ❑ Boston Consulting Group.-
- ❑ Oferta y demanda de empleo cualificado en la Nueva Economía, <http://www.infoempleo.com>
- ❑ Principaux Indicateurs Économiques OCDE.- <http://www.oecd.org>
- ❑ <http://www.comertronic.com>
- ❑ <http://www.e-busines.pwcglobal.com>
- ❑ <http://www.ec-europe.org>
- ❑ <http://www.ecommerce.gov>
- ❑ <http://www.ecommercetimes.com/small-business/>
- ❑ <http://www.intelliquest.com>
- ❑ <http://www.nasscom.org>
- ❑ <http://www.cordis.lu/esprit/arc/ecomint.htm>
- ❑ <http://www.monografias.com>

ANEXO I

ANTECEDENTES DE LOS SISTEMAS DE COMERCIO ELECTRONICO

1.1 ORIGEN

Aunque las primeras iniciativas referentes a la utilización de Internet surgieron a finales de la década de los años 60, en los últimos años es cuando se ha desarrollado con fuerza como un mecanismo alternativo para la realización de actividades económicas.

De 1969 a 1983 (Fig. 1), hubo un lento crecimiento de equipos conectados a la red, sin ninguna manifestación de actividades económicas.

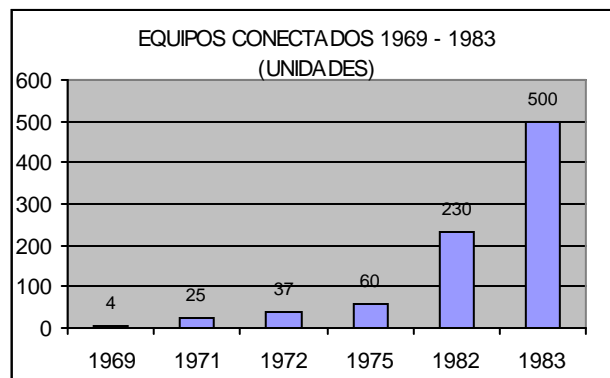


Fig. 1.- Equipos conectados a Internet 1969-1983¹

Hay que tomar en cuenta que el número de equipos conectados a Internet, no indica el número de usuarios que acceden a la misma, ya que existe un factor de proporcionalidad que cambia continuamente y que indica el número de usuarios que se conectan a Internet a través de un mismo equipo.

¹ Según datos obtenidos por una unidad de investigación de Argentina. <http://www.dbinternet.com.ar>.

De 1984 a 1991 (Fig. 2), se da un crecimiento considerable y se avizora un incremento acelerado, por lo que se comienza a difundir la utilización de Internet más allá de los campos militares y científicos.

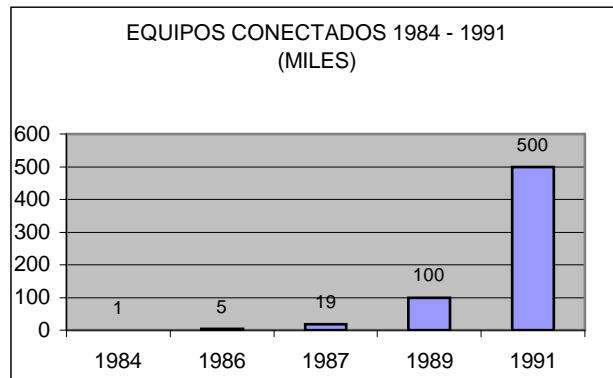


Fig. 2.- Equipos conectados a Internet 1984 a 1991¹

A inicios de la década de los noventa el Internet, comienza a ser utilizado para llevar a cabo actividades económicas, sobresaliendo el comercio electrónico, motivo de esta investigación.

1.2 EVOLUCION

El crecimiento del comercio electrónico no ha sido un fenómeno aislado, sino que se produce como consecuencia de la conjunción de por lo menos tres aspectos fundamentales:

1. Tecnología: Fibra óptica, tecnología digital, satélites, etc.
2. Aceptación de la Sociedad: Importante incremento en el uso de Internet.
3. Asistencia de Organismos Reguladores: Aprobación de leyes que lo avalizan.

Estos tres aspectos permitieron que durante la década de los noventa (Fig. 3), se de un incremento acelerado de equipos en Internet, mejorando las condiciones para poder llevar a cabo transacciones comerciales que cada vez exigen mas sofisticación de esta infraestructura.

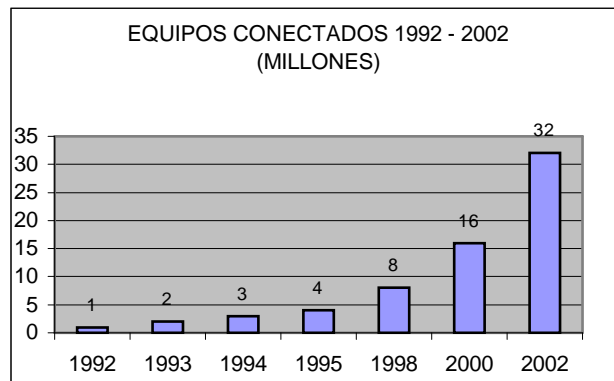


Fig. 3.- Equipos conectados a Internet la última década¹

Desde los inicios del comercio electrónico se pueden identificar tres etapas. Sin embargo, debido a la gran rapidez de los cambios en tecnología, aceptación y adopción del medio, estas etapas se traslapan.

1.2.1 Primera Etapa.- *La era del folleto electrónico.-* Con una cantidad considerable de usuarios y la introducción del primer navegador gráfico, NCSA Mosaic, en 1993, se marcó este inicio. La tesis principal de ésta etapa era la de difundir la imagen y las características más importantes de una empresa y sus productos en una página web estática. La idea principal era encontrarse en Internet, los primeros en incursionar fueron las grandes empresas.

1.2.2 Segunda Etapa.- *La era de la tienda en línea.-* Condiciones tecnológicas y el apareamiento de nuevas herramientas que permitían interacción con sistemas existentes de bases de datos, comunicaciones y sistemas subyacentes (como cobro por tarjeta) inician una campaña para comercializar los productos. La idea preponderante de esta etapa era vender, al tiempo que se buscaba disminuir los costos de venta y atención a clientes. El mayor fenómeno de esta etapa, fue los negocios de Internet puro, empresas creadas desde cero para vender artículos o servicios. Amazon.com es la empresa más conocida mundialmente nacida en esta etapa.

El asombroso crecimiento del comercio electrónico en esta etapa, llegó a su punto más alto en 1997-98, los negocios de Internet estaban en la bolsa de valores NASDAQ, de los Estados Unidos, que es la bolsa mundial más importante para empresas de tecnología, posicionándose Amazon como la empresa más poderosa y representativa del comercio electrónico en Internet puro.

Como se puede anticipar en esta etapa ya nace la necesidad de tener Sistemas de Bases de Datos que interactúen con los sistemas subyacentes de comercio electrónico.

1.2.3 Etapa Actual.- *Maneja el concepto de B2B, business to business.*- El énfasis de esta etapa es utilizar Internet como una parte integral para eficientar los procesos entre las empresas, para aprovechar las ventajas de comunicación, acceso y procesamiento de la información de bases de datos.

Las empresas comienzan a ser conscientes de la nueva realidad, aprovechando esta oportunidad de negocio, tanto desde la óptica del recorte de costes como desde la de nuevos ingresos. Se estima que su volumen de negocio llegará a las cotas similares del comercio tradicional en tan sólo 5 años, reduciendo tres veces el tiempo que el comercio tradicional tardó.

Las transacciones comerciales electrónicas surgió y está probada su efectividad en la virtualización de las relaciones de negocios, obteniendo beneficios como:

- Menores costos operativos: espacio físico, mobiliario, agua, luz, etc.
- Mejor atención al usuario: a disposición las 24 horas del día, los 365 días del año.
- Mayor cobertura: a escala mundial.
- Mayor rentabilidad de sus negocios: menor inversión y mayor rendimiento.

Por el lado del consumidor, este tiene beneficios como:

- Comodidad: al realizarlo desde su computadora en su casa u oficina.

- Rapidez: no desperdicia tiempo en trasladarse hasta el proveedor, negociación, etc.
- Mejor toma de decisión: Puede comparar mayor número de ofertas en menor tiempo

En esta etapa los participantes ya no son solo empresas de élite tecnológica como Intel, Microsoft, General Motors, etc. Muchas compañías pequeñas están adoptando procesos de negocios que las grandes empresas emplean en el Intercambio Electrónico de Datos (EDI).

1.3 DIFERENCIAS SECTORIALES

Evaluar la posibilidad de comerciar a través de Internet, implica disponer de un profundo conocimiento de bases de datos, así como de las peculiaridades que atañen a cada sector. Para poder determinar la existencia de oportunidades reales se deben analizar cinco aspectos:

1. *Atractivo del Mercado.*- Se analiza el tamaño actual y futuro del mercado on-line. De esta forma se estiman los ingresos potenciales del mercado en cuestión.
2. *Nivel de Fragmentación o Concentración Empresarial.*- Se considera este aspecto para poder determinar el número de miembros que pueden operar a través de un mercado.
3. *Situación de los Intermediarios.*- Se analiza a los elementos que intervienen en la distribución y suministro de productos en cada sector.
4. *Adaptación al Medio On-line.*- Factores como la innovación tecnológica, marco legal y el nivel de utilización de Internet en un determinado sector contribuyen a modificar un determinado mercado.
5. *Entorno competitivo.*- Comprueba si la oportunidad de mercado ya ha sido ocupada, o caso contrario, determinar cuanto mercado queda aún por ocupar.

1.3.1 MUNDIAL

Según la firma investigadora de mercados Nielsen a finales del 2002, el número de navegantes en el mundo era de unos 500 millones, de los cuales 498 millones acceden a la red desde sus hogares. Para

tener una clara visión es necesario analizar retrospectivamente el incremento de usuarios en Internet a partir de la aparición del comercio electrónico. Como se puede ver (Fig. 4), el crecimiento ha sido abrumador, con claro predominio de Norteamérica.

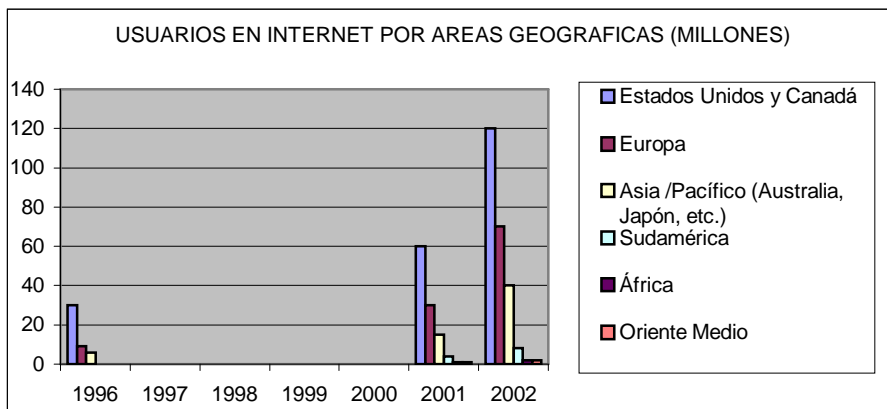


Fig. 4.- Distribución de usuarios de Internet por áreas geográficas²

En similar análisis, se nota un crecimiento considerable de las transacciones comerciales virtuales.

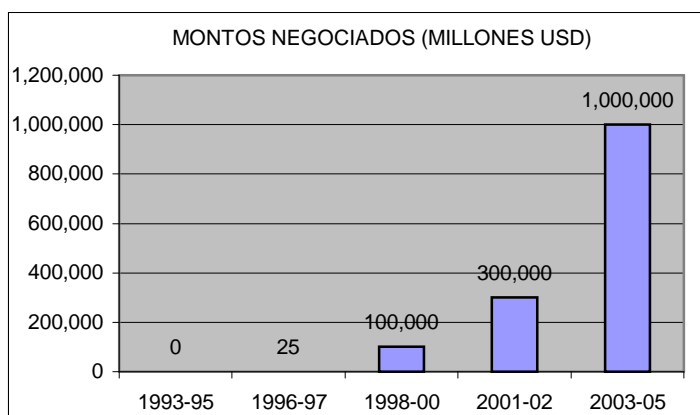


Fig. 5.- Volumen de comercio electrónico³

Con estos volúmenes comerciales se puede justificar, por qué incluso los países considerados en vías de desarrollo están invirtiendo para poder aprovechar las ventajas de las transacciones comerciales en Internet⁴. En dicho análisis se valoran 70 indicadores distintos.

² Según la firma investigadora de mercados Nielsen

³ Según informe de la OCDE sobre el Impacto Social y Económico del Comercio Electrónico (2000)

1.3.2 REGIONAL

Según informe de la OCDE sobre el Impacto Social y Económico del Comercio Electrónico 2002, en Sudamérica estima que existen 8 millones de usuarios de Internet. Según otras consultoras las tasas de crecimiento en los próximos años son muy importantes, según eMarketer a finales del 2003 se habrá alcanzado los 19,28 millones de usuarios, mientras que para Júpiter Communications la cifra podría llegar a los 38 millones de usuarios⁵.

Si bien es cierto que este número implica solamente el 3.7% de la población, los expertos indican que se trata del segmento con mayor poder de adquisición, que tienen acceso a tarjetas de crédito y que viven en zonas donde la distribución y entrega de productos no es difícil. Si a esto sumamos que están desapareciendo, cada vez más, los obstáculos materiales del comercio electrónico en América Latina (Fig. 6), la conclusión final es que muy pronto el comercio electrónico será parte de la vida de todos nosotros, independientemente del lugar del mundo en el que nos encontremos.

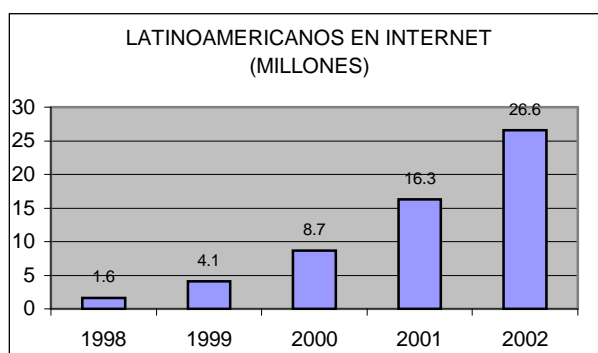


Fig. 6.- Millones de latinoamericanos en Internet¹

La expansión de Internet en la región tiene como principal muestra el acelerado crecimiento en cuanto al número de personas conectadas. Según IDC para la región tiene proyectado un crecimiento anual

⁴ Según informe elaborado por The Economist Intelligence Unit (EIU), ranking de los 60 mas países preparados para el comercio electrónico a nivel mundial. <http://www.ebusinessforum.com>

⁵ Véase <http://www.commerce.net>

del 32%, el más elevado del mundo. Según el *Wall Street Journal*, en otra investigación realizada en junio 2002, el total del comercio electrónico en Latinoamérica asciende a \$ 1.060 millones de dólares, y pese a que más de la mitad de los internautas son brasileños, los más entusiastas frente al comercio electrónico son los panameños, ecuatorianos y venezolanos en un 58%, 55% y 53%, respectivamente.

Si hace unos años hablar de comercio electrónico y economía digital parecía un tema relacionado exclusivamente con las áreas más desarrolladas del planeta, en la actualidad el panorama ha cambiado considerablemente. Muchos países en vías de desarrollo como Brasil, Argentina, Ecuador, Venezuela, Perú y Colombia están optando por la utilización de la tecnología de la información como alternativa para potenciar su economía y lograr mejorar su situación económica internacional en el futuro. El mercado latinoamericano esta ahora mismo en plena expansión, que si en 1999 generó 200 millones de dólares se espera que para el 2005 la cifra llegue a los 8.300 millones de dólares⁶, lo que hace al mercado latinoamericano muy interesante para las empresas de comercio electrónico a nivel mundial.

1.3.3 NACIONAL

Según la Corporación Ecuatoriana de Comercio Electrónico (CORPECE) y la Superintendencia de Telecomunicaciones, en agosto del 2000 existía 48.535 usuarios de Internet registrados, mientras que para agosto del 2001, esta cifra ascendió a 78.527 usuarios del Internet en el Ecuador registrados. Hay que aclarar que se trata de una cifra irreal, toda vez que dentro de los inscritos generalmente se cuentan personas jurídicas que engloban cientos y hasta miles de usuarios bajo una sola inscripción. Lo importante es mencionar que si bien las cifras pueden no ser exactas, las perspectivas siempre van en una progresión ascendente.

A la vista de los resultados existentes en países desarrollados, especialmente EEUU, la Unión Europea y Japón, nuestro país esta intentado potenciar con fuerza la economía digital, pues, los diferentes sectores están preparando el escenario adecuado. Entre estas podemos destacar:

⁶ Según consultora Jupiter Communications, <http://www.ebusinessforum.com>

- Estado: La ley de comercio electrónico aprobada por el congreso el 31 de Diciembre de 2002 Ley que avaliza transacciones comerciales en Internet.
- Comunicaciones: Nuevas bandas de telefonía, así como la troncal de fibra óptica en desarrollo en algunos sectores del país.
- Sector privado: Empresas que invierten en tecnología que permita mostrarse en Internet y para poder poner a disposición sus productos y servicios on-line.

Con todos estos antecedentes de crecimiento de equipos, usuarios y volumen de transacciones, es imprescindible analizar el comportamiento del sistema de comercio electrónico cuando el factor más determinante en su éxito depende del nivel de rendimiento del mismo.

1.4 FUTURO DE LOS SISTEMAS E-COMMERCE

Muchos países están potenciando el desarrollo del comercio electrónico tras observar los beneficios que ha tenido el líder indiscutible, Estados Unidos, que controla el 80% a nivel mundial y el 20% del comercio electrónico de Europa que es la segunda potencia a nivel mundial en el desarrollo de este tipo de comercio.

El crecimiento de usuarios de Internet en Ecuador y la evolución de las operaciones económicas realizadas a través de él advierte un excelente futuro, sin embargo, conviene tener presente que inherentes a él hay una serie de temas que se deben considerar. Esto es que el comercio electrónico no se está desarrollando en todos los países por igual ya que algunos tienen una clara ventaja en esta cuestión⁷, como comunicaciones, legislación, cultura etc. Sin embargo, podemos afirmar que hoy en día Ecuador esta impulsando fuertemente el comercio electrónico, por lo que el proceso de consolidación en la sociedad ecuatoriana se esta desarrollando rápidamente.

⁷ Según datos de la Boston Consulting Group

1.5 INTEROPERABILIDAD Y ESTANDARES EN SERVICIOS VIA WEB

Uno de los problemas que abundan en Internet es la falta de estándares de Interoperabilidad y estándares en Servicios Vía Web definidos. Sin embargo, cuando estos estándares son definidos y respetados por las empresas del sector, aparece una con el peso suficiente como AOL o Microsoft para estropear iniciativas que serían beneficiosas para todos excepto para sus respectivos monopolios.

La última tendencia en Internet son los WebServices, por primera vez en bastante tiempo una iniciativa conjunta de IBM y Microsoft, crearon una fundación con el objetivo de definir estándares de interoperabilidad para servicios vía Web llamada *WebServices Interoperability Organization*. Esta Organización tiene como objetivo: "*promover entre la industria, la interoperabilidad de los Web Services a través de cualquier plataforma, sistemas operativos y lenguajes de programación y proveer recursos, guías y best-practices para desarrollar soluciones de Web Services*".

ANEXO II

REGLAS COMERCIALES DE COMERCIO ELECTRONICO

Como se indica en el Capítulo III, estas permiten resaltar las peculiaridades de sistemas de esta naturaleza. Las respuestas a este cuestionario proporcionan útil información a tener cuenta en el momento del desarrollo.

Presentación de productos y servicios:

- Cuántas líneas diferentes de productos va ha manejar? Ej. Electrodomésticos y/o equipos deportivos.
- Cuál es el rango en cada una de las líneas? Ej. En electrodomésticos una o varias marcas.
- Los productos son tangibles o intangibles? Ej. Un televisor o acuerdos de servicios.
- Qué mecanismo se utiliza para la entrega de los productos? Ej. Vía correo convencional (televisor) o vía correo electrónico (acuerdo de servicios).
- La frecuencia con que los productos y/o su información cambia? Ej. Modelos de portátiles.
- Como se desplegara la información de los productos? Texto, imagen, video, etc.

Categorización de precios:

- Todos los artículos tiene un solo precio?
- Los precios de los artículos se categoriza de acuerdo a los clientes?
- Maneja precios por promoción, temporada, etc. para los productos?
- Qué precios maneja para los canales de distribución?

Incentivos por compras:

- Se usa cupones de compra o membresías?
- Aplica incentivos en las compras?
- Se realiza descuentos en compras por cantidad o volumen?

Accesorios compatibles y de sustitución

- Brinda información a cerca de los accesorios compatibles o de reemplazo a los artículos que promociona?
- Brinda asistencia en caso de reemplazo o sustitución?

Categorización y subcategorización de ítems

- Ayuda en la búsqueda de ítems a través de categorización y subcategorización de los mismos?
- Cuántas categorías de ítems tiene su empresa?
- De que forma categorizar los productos?
- Cuál es la forma que se conocen comercialmente los productos?

Herramientas de búsqueda:

- Desea implementar búsquedas por palabra clave?
- A que detalle quiere implementar la búsqueda? Ej. Marca, modelo, descripción, etc.
- Las herramientas incluye búsquedas por rango?.

Definición de configuraciones:

- Las ordenes de compra pueden ser personalizadas?
- Los productos pueden ser configurados para armar un equipo? Ej. Computador P4...

Proporciona asesoría en las ventas:

- Asesora en las alternativas de soluciones? Ej. Plan de telefónica celular corporativo.

- Brinda información comparativa a cerca de las soluciones que ofrece?

Servicio de Preguntas frecuentes:

- Se proporcionará un canal para responder a las inquietudes de los clientes?
- De que forma manejará las contestaciones?

Manejo y procesamiento de transacciones:

- Cuál será el procedimiento para realizar la compra?
- De cuantos ítems esta compuesta una orden en general?
- Si permite ordenes en espera y como se contabiliza los productos de ella?
- En que tiempo se procesa una transacción? Tiempo real, cada hora, cada día, etc.

Impuestos y envío

- Los ítems incluye impuestos o se desglosa al final del pedido?
- Los gastos de envío se incluyen en los ítems es cubierto por el cliente?
- En el caso de que se incluya gastos de transporte. Cómo se estima su costo?
- Qué opciones de envío proporcionara?

Impuestos de destino

- Qué impuestos son aplicados a cada cliente? Ej. Instituciones, ONG, tercera edad.
- Qué impuestos son aplicados a cada ítem? Ej. Bebidas (ICE), insumos agrícolas.

Tipos de Pago

- Qué formas de pago están disponibles? Tarjeta de crédito, deposito, debito.
- Cuánto tiempo se demora en procesar los pagos?

Autenticación, validación y aclaración de la transacción

- Qué nivel de autenticación se requiere? Aunque una rigurosa autenticación disminuye el riesgo, aumenta la dificultad del proceso de pago, que puede llevar a disminuir las probabilidades de concretar una venta.
- Cuáles serán las políticas a seguir?
- Hasta que nivel de riesgo de fraude potenciales se está protegiendo?
- A qué información se aplicarán algoritmos de autenticación? Número de tarjeta de crédito, número de identificación personal, etc.

Integración con el inventario

- Solo se presentarán artículos que se dispone en inventarios?
- Se presentarán todos los artículos incluidos, aun los que no se dispone para entrega inmediata?

Controles

- Cuándo se dejan de ofrecer los artículos para la venta? Cuando se maneja un mínimo de existencias, se sobrepasa este límite ó cuando no hay existencias.
- La política de oferta es para todos los productos o es diferente para cada producto?

Procesamiento de las órdenes

- Cada que tiempo se procesan las ordenes? En tiempo real, cada hora, una vez al día.
- Cuáles son las políticas de despacho de productos? Cuando se tiene acreditado el valor, cuando llega el depósito, etc.
- Hasta que hora se puede despachar los productos? Puede darse el caso de que la compra es hecha al final de la jornada laboral.

Reverso de las órdenes

- Cuáles son las condiciones para reversar una orden?
- En que momento se notificará al cliente cuando una orden se reversa?

Servicio al cliente

- El sistema proporcionara información a cerca del estado de la orden?
- Permite saber en que situación del envío se encuentra los productos?
- Posee conexión con el encargado del envío? UPS, FedEx, or USPS tracking numbers.

Notificaciones del estado de la orden

- Se utilizará un mecanismo adicional para mantener contacto a cerca del estado de la orden?
Mail, teléfono.
- Se pedirá una aceptación de la orden antes de tramitarla?

Soporte técnico

- Se incorporará material de apoyo a los productos que se ofertan?
- Cuáles son las condiciones para el soporte técnico?
- Necesita ser procesado como una orden?

Localización

- Necesita manejar varios idiomas? Ingles para EEUU, español para América Latina, etc.
- Se manejará una o varias unidades monetarias?
- Los precios tienen alguna relación con la región?
- Hay versiones de los productos por regiones?

Situación Legal

- En que políticas legales esta amparado el cliente?

- En que políticas legales esta amparada la empresa?
- Existe contradicción entre los amparos legales del cliente y la empresa?
- Es necesario llenar un acuerdo comercial con el cliente?
- Se contempla las políticas de garantías y rescisión?
- En caso de devoluciones como se manejarán esos costos?

Manejo de las expectativas del cliente

- Qué expectativas de cumplimiento desea crear en sus clientes?
- Qué tiempos se manejará para cada etapa?

Obtención de métricas y datos

- Métricas del site
 - Qué métricas se usará para el site?
 - Qué informes estarán disponibles para el cliente, y cuales para la empresa? Clientes, ordenes, estado de las ordenes, análisis del tráfico en el site.
 - Cuáles son las tácticas por las cuales se incrementó o cayó las ventas? y por qué?
 - Cómo se medirá tácticas de anuncios y mercadeo directo implementadas?
 - Cómo se evaluará y manejará la satisfacción del cliente?
- Perfil del cliente
 - Cuál y como se obtendrá información adicional?
 - Qué incentivos se utilizarán para que los usuarios anónimos se registren?
 - Qué datos demográficos desea obtener en el registro de clientes?

Afinamiento de las tácticas de mercadeo

- Datamining
 - Qué información de sus clientes tiene registrado?

- Cómo se utilizará la información para optimizar las tácticas de mercadeo?
- Desea ayudar en las futuras compras de acuerdo al historial de las anteriores?
- Qué hábitos de búsqueda y elaboración de ordenes utilizan los clientes?
- Demográficos
 - Cuáles son los clientes?
 - Se pueden clasificar a los clientes?
 - Qué tanto conocen los clientes acerca de los productos?
 - Cuánta ayuda necesitarán para decidirse?
 - En qué porcentaje los clientes son personas individuales o compañías?
 - Qué tan frecuente visitan los clientes el sitio?
 - Qué tan frecuente las visitas se concretan en un pedido?

Categorización de clientes

- Existe categorización de clientes?
- Los productos están disponibles para ciertas categorías de clientes?
- Se han agrupado los productos de acuerdo a los clientes?

Promoción del sitio

- Cómo avisará que su negocio está presente en Internet?
- Se registrará en máquinas de búsqueda y/o portales de Internet el sitio Web?
- Proporcionará publicidad a otros portales?
- Se manejan tácticas de mercadeo directas para clientes potenciales?
- Se utilizarán relaciones públicas o grupo de noticias?
- Se utilizarán promociones, concursos o ofertas?

Análisis de ventas

- Cuáles son las razones por las cuales los clientes lo seleccionaron para realizar sus compras?
Precio, variedad, servicio, agilidad, disponibilidad, etc.
- Qué se ofrece a los clientes para ser competitivo?

Administración del software y hosting

- Como va ha estar relacionada la empresa con la administración del sitio?
- Se desea y/o necesita controlar el servidor las 24 horas y los 7 días de la semana?
- Se desea y/o necesita que la administración del sitio del hosting este en manos de terceros?

Utilización de la aplicación

- Exploración de la base de datos
 - Cuál será el método de exploración adecuado? Al que esta acostumbrado el cliente, mecanismo de configuración de navegación, etc.
 - Qué facilidades de exploración brindan las diferentes tecnologías que utilizan el software de las bases de datos?
 - Cuál es el flujo de la información cuando se dan consultas exigentes?
 - Hasta que punto es aceptable el tiempo de recuperación y/o exploración?
- Navegación por la aplicación
 - Cuáles son los hábitos de navegación de los clientes?
 - Se rastreará el origen de las ventas y el recorrido hasta que el cliente concreta el pedido?
 - Los archivos .log brindan valiosa información a cerca de los hábitos de navegación del cliente. Hasta que tamaño se manejarán dichos archivos?
 - En que porcentaje el software utilizado es compatible con el del usuario?

Varios

- Cuál es la manera mas rápida de hacer compras?
- Cuáles son los patrones de navegación de los clientes?
- Cuáles son los servicios que le hacen único en el mercado?
- Cuál es la diferencia con la competencia?

Cómo comunicar las fortalezas del negocio a sus clientes?

ANEXO III

CODIGO DE LA APLICACIÓN WEB DE COMERCIO ELECTRONICO

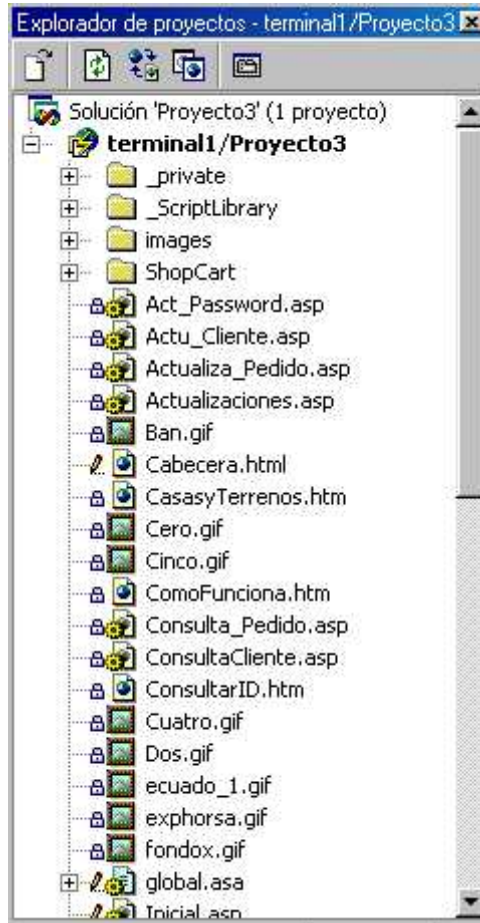


Fig.- Ventana de exploración de proyectos

A continuación se presentan los archivos que se considera relevantes del sistema de comercio electrónico. Cabe recalcar que se han remplazado algunas líneas por ser valores conocidos, como por ejemplo en ves de enumerar todas las provincias se ha sustituido por “Provincias”. Esto con el fin de que no extender esa presentación con información trivial.

Archivo: PagInicial.htm

```

<html>
<head>
<meta>
<title>COMPUTEC - ECUADOR </title>
</head>
  <frameset rows="15%,*">
    <frame name="Cabecera" src="Cabecera.html" scrolling="none" frameborder="0"
      FRAMESPACING="0" scrolling="off" noresize border="0">
    <frameset cols="15%,*,15%">
      <frame name="Menu" src="Menu.htm" scrolling="auto" frameborder="0"
        FRAMESPACING="0" scrolling="off" noresize border="0">
      <frame name="Cuerpo" src="Inicial.asp" scrolling="auto" frameborder="0"
        FRAMESPACING="0" scrolling="off" noresize border="0">
      <frame name="Publicidad" src="Publicidad.htm" scrolling="auto"
        frameborder="0" FRAMESPACING="0" scrolling="off" noresize
        border="0">
    </frameset>
  </frameset>
</html>

```

Archivo: Inicial.asp

```

<% @ Language=VBScript %>
<% Option Explicit %>
<html>
<head>
<title>Comercial Puruhá</title>
<meta name="Microsoft Theme" content="expeditn 011"><meta name="Microsoft Border"
content="none, default"></head>
<body bgcolor="#ffffff" text="#000000" link="#006600" vlink="#669966" alink="#990000"
topMargin=5 leftMargin=5 rightMargin=5 bottomMargin=5>
<form name="FormularioPass" action="PassWord.asp" title="PassWord" onsubmit="return
Validate();">
  <script Language="JavaScript">
    function Validate()
    {
      if ((document.FormularioPass.ID.value == "
)|(document.FormularioPass.TxtContrasena.value == " ))
      {
        alert("Necesitamos su Alias y su Número de Cliente")
        return false;
      }
      return true;
    }
  </script>
<center><IMG height=13 src="sep.gif" width=493></center>
<P align=center>
<font face="Verdana"><strong>
Bienvenidos a Computec.com<br>
<center>Para iniciar su Pedido ingrese sus datos.</center><br><br>
<center>

```

```

<table>
  <TR>
    <td align=right><b>Contraseña:</b></td>
    <td align=left><b><input type=password name=TxtContrasena id=TxtContrasena
  ></b></td>
  </TR>
  <TR>
    <td align=right><b>Número de Cliente:</b></td>
    <td align=left><b><input name=ID id=ID
  ></b></td>
  </TR>
</table>
</center>
<center><input type="submit" value="Continuar" id="submit1" name="submit1"><br></center>
<br>
<P align=center>
Si todavía no tiene un Número de Cliente por favor <A href="Registro.htm">regístrese</A> o si se
olvidó su Número de Cliente consúltelo <A href="ConsultarID.htm">
aquí.</A></strong></font></P>
<P></P>
<P></P></form>
<br>
<center><IMG height=10 src="exphorsa.gif" width=300></center>
<br>
<p align="center"></FONT><strong><font face="Verdana">Este es un prototipo de sistema de
comercio
electrónico diseñado para demostrar la interoperabilidad con una base de datos
Centralizada.</BIG></font></FONT></FONT> </strong></p>
<p></p>
<P align=center><font face="Verdana" size="4"><strong><font size="3">

Marco V. Yambay D.<br><br>

</font></strong></font></P>
<center>
<table>
  <tr><td><center><font face=verdana size=2><b>Ud. es el visitante
Nº:</b></font></center></td></tr>
  <tr>
    <td>
      <center>
        <%
          Dim RSVisitas, DBConn
          Dim nVisitas
          Dim Uni, Dec, Cen, Mil, DMil, Aux
          set DBConn=server.CreateObject ("ADODB.Connection")
          DBConn.Open "ConPuruha"
          set RSVisitas=server.CreateObject ("ADODB.Recordset")
          set RSVisitas=DBConn.Execute ("Select Num_Visitante from
          Usuario")

```

```

nVisitas=RSVisitas("Num_Visitante")
nVisitas=nVisitas*1
nVisitas=nVisitas + 1
DBConn.Execute ("Update Usuario set Num_Visitante=" & nVisitas)
nVisitas=nVisitas - 1
Aux=nVisitas
if Aux>=1 and Aux<=9 then
    Response.Write("<IMG height=20 src=Cero.gif width=15>")
    Response.Write("<IMG height=20 src=Cero.gif width=15>")
    Response.Write("<IMG height=20 src=Cero.gif width=15>")
    Response.Write("<IMG height=20 src=Cero.gif width=15>")
    select case Aux
        case n: "contador de visitas"
    end select
end if

'Fin Miles
%></center>
</td>
</tr>
</table>
</center>

<P align=center><FONT face=verdana size=1><STRONG>
Contáctenos y envíenos sus preguntas y
comentarios a:<br>
<br>
<A href="mailto:informacion@computeec.com">informacion@computeec.com</A><br>
<br><A href="mailto:comercial.puruha@comercialpuruha.com">Riobamba:</A>(593-3)- 2944-223
<br></STRONG></FONT></P>
<center><IMG height=13 src="sep.gif" width=493></center>

</body>
</html>

```

Archivo: Registro.asp

```

<html>
<head>
<meta>
<title></title>
</head>
<Script Language=JavaScript>
    function Validate(){
        //Vacío
        if (document.FrmClientes.TxtCedula.value == "")
        {
            alert("La cédula no puede ser un Campo Vacío")
            return false
        }
        //Números

```

```

if (isNaN(document.FrmClientes.TxtCedula.value)){
    alert("Necesitamos Números para la Cédula")
    return false;
}
//Números
if (isNaN(document.FrmClientes.TxtPasaporte.value ))
{
    alert ("Necesitamos Números para el Pasaporte")
    return false;
}
//Vacío
if (document.FrmClientes.TxtApellidos.value=="")
{
    alert ("Necesitamos sus Apellidos")
    return false;
}
//Vacío
if (document.FrmClientes.TxtNombre.value=="")
{
    alert ("Necesitamos sus Nombres")
    return false;
}
if (document.FrmClientes.TxtContrasena.value=="")
{
    alert ("Necesitamos una Contraseña para autenticarlo")
    return false;
}
if (document.FrmClientes.TxtProvinciaEstadoExter.value=="")
{
    alert ("Necesitamos el Estado o Provicnia en donde se encuentra")
    return false;
}
if (document.FrmClientes.TxtCiudadExterior.value=="")
{
    alert ("Necesitamos la Ciudad en donde se encuentra")
    return false;
}
if (document.FrmClientes.TxtDireccionExterior.value=="")
{
    alert ("Necesitamos la Dirección en donde se encuentra")
    return false;
}

return true;
var WGdc=","; var WGgc=","; var WGnc="("; var WGcs=" ";
}

```

</Script>

<body background ="" bottomMargin="5" leftMargin="5" rightMargin="5" topMargin="5">

<center></center>

<center>Registro de Clientes de

```

Computeec.com</strong></font></center>
<form name="FrmClientes" action="Registro.asp" onsubmit="return Validate();">
<center>
<table>
  <tr>
    <td align="right">
      <p align="right"><font face="Verdana" size="2">Cédula Ecuatoriana: </font> </p> </td>
    <td>
      <p align="left"><font face="Verdana" size="2"><input id="TxtCedula"
      Name="TxtCedula"></font></p></td>
    </tr>
    <tr>
    <td align="right">
      <p align="right"><font face="Verdana" size="2">Pasaporte:</font></p></td><td>
      <p align="left"><font face="Verdana" size="2"><input id="TxtPasaporte"
      name="TxtPasaporte"></font></p></td>
    </tr>
    <tr>
    <td align="right">
      <p align="right"><font face="Verdana" size="2">Apellidos:</font></p></td>
    <td>
      <p align="left"><font face="Verdana" size="2"><input id="txtApellidos"
      name="TxtApellidos"></font></p></td>
    </tr>
    <tr>
    <td align="right">
      <p align="right"><font face="Verdana" size="2">Nombres:</font></p></td>
    <td>
      <p align="left"><font face="Verdana" size="2"><input id="TxtNombre"
      name="TxtNombres"></font></p></td>
    </tr>
    <tr>
    <td align="right">
      <p align="right"><font face="Verdana" size="2">Contraseña:</font></p></td>
    <td>
      <p align="left"><font face="Verdana" size="2"><input id="TxtContrasena"
      name="TxtContrasena"></font></p></td>
    </tr>
    <tr>
    <td align="right">
      <p align="right"><font face="Verdana" size="2">email:</font></p></td>
    <td>
      <p align="left"><font face="Verdana" size="2"><input id="Txtemail"
      name="Txtemail"></font></p></td>
    </tr>
    <tr>
    <td align="right">
      <p align="right"><font face="Verdana" size="2">Fecha de
      Nacimiento:</font> </p> </td>
    <td>
      <p align="left"><select id="SelDia" name="SelDia" style="WIDTH: 38px; HEIGHT: 22px">
      <option selected>1
      “selección de opciones”

```

```

</option>
</select><font face="Verdana" size="2">
</font>
<select id="SelMes" name="SelMes" style="WIDTH: 94px; HEIGHT: 22px">
<option selected>
“meses”
</option>
</select><font face="Verdana" size="2">
</font>

<select id="SelAnio" name="SelAnio" style="WIDTH: 53px; HEIGHT: 22px">
<option selected>1940
“año”
</option>
</select><font face="Verdana" size="2">
</font> </p>
</td>
</tr>
</tr>
<td align="right">
<p align="right"><font face="Verdana" size="2">Provincia en Ecuador:</font> </p> </td><td>
<p align="left"><select id="SelProvinciaenEcuador" name="SelProvinciaenEcuador" style="WIDTH:
94px; HEIGHT: 22px">
<option selected>
“Provincia”
</option>
</select></p></td>
</tr>
</tr>
<td align="right">
<p align="right"><font face="Verdana" size="2">
Ciudad en Ecuador:
</font> </p> </td> <td>
<p align="left"><input id="TxtCiudadenEcuador" name="TxtCiudadenEcuador"></p>
</td>
</tr>
</tr>
<td align="right">
<p align="right"><font face="Verdana" size="2">Dirección en
Ecuador:</font> </p> </td>
<td>
<p align="left"><font face="Verdana" size="2"><input id="TxtDireccionenEcuador"
name="TxtDireccionenEcuador"></font></p></td>
</tr>
</tr>
<td align="right">
<p align="right"><font face="Verdana" size="2">Teléfono en
Ecuador:</font> </p> </td>
<td>
<p align="left"><font face="Verdana" size="2"><input id="TxtTelefonoenEcuador"
name="TxtTelefonoenEcuador" maxLength="8" style="WIDTH: 76px; HEIGHT:
22px"></font></p></td>
</tr>
</table></center>

```

```

        <div align="center"><font face="Verdana" size="2">Datos donde se encuentra</font>
        </div>
<div align="center">
    <table>
        <tr>
            <td align="right">
                <p align="right"><font face="Verdana" size="2">País en donde
                está:</font> </p> </td>
                <td><select id="SelPais" name="SelPais" style="WIDTH: 155px; HEIGHT: 22px">
                    <option selected>Alemania

                    <option>
                    “pais”
                </option>
                </select></td>
            </tr>
            <tr>
                <td align="right"><font face="Verdana" size="2">Provincia /
                Estado:</font> </td>
                <td><font face="Verdana" size="2"><input id="TxtProvinciaEstadoExter"
                name="TxtProvinciaEstadoExter"></font></td>
            </tr>
            <tr>
                <td align="right"><font face="Verdana" size="2">Ciudad:</font></td>
                <td><font face="Verdana" size="2"><input id="TxtCiudadExterior"
                name="TxtCiudadExterior"></font></td>
            </tr>
            <tr>
                <td align="right"><font face="Verdana" size="2">Dirección:</font></td>
                <td><font face="Verdana" size="2"><input id="TxtDireccionExterior"
                name="TxtDireccionExterior"></font></td>
            </tr>
            <tr>
                <td align="right"><font face="Verdana" size="2">Código
                Postal:</font> </td>
                <td><font face="Verdana" size="2"><input id="TxtCodigoPostalExter"
                name="TxtCodigoPostalExter"></font></td>
            </tr>
            <tr>
                <td align="right"><font face="Verdana" size="2">Teléfono:</font></td>
                <td><font face="Verdana" size="2"><input id="TxtTelefonoExterior"
                name="TxtTelefonoExterior" style="WIDTH: 67px; HEIGHT: 22px"
                ></font></td>
            </tr>
        </table></div>
<div align="center"><font face="Verdana" size="2">Datos de la
Persona que recibe</font> </div>
<center>
    <table>
        <tr>
            <td align="right">
                <p align="right"><font face="Verdana" size="2">Apellidos:</font></p></td>
            <td>

```



```

    <p align="left"><font face="Verdana" size="2"><input id="TxtApellidosRecibe"
name="TxtApellidosRecibe"></font></p></td>
    </tr>
    <tr>
        <td align="right">
            <p align="right"><font face="Verdana" size="2">Nombres:</font></p></td>
        <td>
            <p align="left"><font face="Verdana" size="2"><input id="TxtNombresRecibe"
name="TxtNombresRecibe"></font></p></td>
        </tr>
    <tr>
        <td align="right">
            <p align="right"><font face="Verdana" size="2">Provincia:</font></p></td>
        <td>
            <p align="left"><select id="SelProvinciaRecibe" name="SelProvinciaRecibe" style="WIDTH:
94px; HEIGHT: 22px">
                <option selected>
                    Provincia
                </option>
            </select></p>
        </td>
    </tr>
    <tr>
        <td align="right">
            <p align="right"><font face="Verdana" size="2">Ciudad:</font></p></td>
        <td>
            <p align="left"><input id="TxtCiudadRecibe" name="TxtCiudadRecibe"></p>
        </td>
    </tr>
    <tr>
        <td align="right">
            <p align="right"><font face="Verdana" size="2">Barrio:</font></p></td>
        <td>
            <p align="left"><font face="Verdana" size="2"><input id="TxtBarrioRecibe"
name="TxtBarrioRecibe"></font></p></td>
        </tr>
    <tr>
        <td align="right">
            <p align="right"><font face="Verdana" size="2">Dirección:</font></p></td>
        <td>
            <p align="left"><font face="Verdana" size="2"><input id="TxtDireccionRecibe"
name="TxtDireccionRecibe"></font></p></td>
        </tr>
    <tr>
        <td align="right">
            <p align="right"><font face="Verdana" size="2">Teléfono:</font></p></td>
        <td>
            <p align="left"><font face="Verdana" size="2"><input id="TxtTelefonoRecibe"
name="TxtTelefonoRecibe"
                style="WIDTH: 77px; HEIGHT: 22px"
            ></font></p></td>
        </tr>
</table></center>

```

```

<br><center><input type="submit" value="Suscribirse" id="ButEnviar" name="ButEnviar">
</center>
</form>
</body>
</html>

```

Archivo: ConsultaID.htm

```

<html>
<head>
<meta>
</head>
<script language=JScript>
    function Validar()
    {
        if ((document.FrmConsulta.TxtCedula.value == "") &&
(document.FrmConsulta.TxtPasaporte.value == ""))
        {
            alert ("Necesitamos uno de los dos datos.")
            return false;
        }
        return true;
    }
</script>
<body topMargin="5" leftMargin="5" rightMargin="5" bottomMargin="5">
<center></center>
<form name="FrmConsulta" Action="ConsultaCliente.asp" onsubmit="return Validar();">
<center><font face="verdana" size="3"><b>Es necesario que ingrese su número de Cédula o
Pasaporte:</b></font></center>
<center><table>
    <tr>
        <td align="Right"><font face="verdana" size="2">Cédula Ecuatoriana:</font></td>
        <td><font face="verdana" size="2"><input type="Text"
name="TxtCedula"></font></td>
    </tr>
    <tr>
        <td align="Right"><font face="verdana" size="2">Pasaporte:</font></td>
        <td><font face="verdana" size="2"><input type="Text"
name="TxtPasaporte"></font></td>
    </tr>
</table>
</center>
<center><br><input Type="Submit" name="BtnConsulta" Value="Consultar"></center>
</form>
</body>
</html>

```

Archivo: ConsultarPedido.asp

```

<% @ Language=VBScript %>
<% Option Explicit%>

```

```

<html>
<head>
<meta>
</head>
<body>
  <SCRIPT LANGUAGE=javascript RUNAT="Server">

    var WGdc=","; var WGgc=","; var WGnc="("; var WGcs=" ";

    function FormatMoney(A,W) {
      var N=Math.abs(Math.round(A*100));
      var S=((N<10)?"00":((N<100)?"0":""))+N;
      S=WGcs+((A<0)?WGnc:"")+WGgroup(S.substring(0,(S.length-
2)))+WGdc+S.substring((S.length-2),S.length)+((A<0&&WGnc=="(")?")":"");
      return (S.length>W)?"Over":S;
    }

    function WGgroup(S) {
      return (S.length<4)?S:(WGgroup(S.substring(0,S.length-
3))+WGgc+S.substring(S.length-3,S.length));
    }

  </SCRIPT>
  <center></center>
  <%
    Dim DBConn, RSPedido, RSPedidoDet, RSCliente
    Dim strConsulta, strConsultaDet, strRecibe
    Dim nCli_Codigo, nPed_Codigo, nSubTotal, nPrecioUn

    Response.Write ("<center><font face=verdana size=3><b>Inicio : " & Timer &
"</b></font></center>")

    nSubTotal=0
    nPed_Codigo=Request.QueryString ("TxtPed_Codigo")
    nCli_Codigo=Request.QueryString ("TxtCli_Codigo")
    if nCli_Codigo="" or nPed_Codigo="" then
      Response.Write ("<br>")
      Response.Write ("<center><font face=verdana size=2><b>Lo sentimos !
Necesitamos necesariamente los dos datos.</b></font></center>")
      Response.Write ("<br>")
      Response.Write ("<center><font face=verdana size=1><b><a
href=Ver_Pedidos.htm>Continuar</a></b></font></center>")
    else
      set DBConn=Server.CreateObject ("ADODB.Connection")
      DBConn.Open "ConPuruha"
      if isnumeric(nCli_Codigo) and isnumeric(nPed_Codigo) then
        strConsulta="Select Ped_Codigo, Pedidos.Cli_Codigo, Cli_Apellidos,
Cli_Nombres, Ped_Fecha, Ped_SubTotal, Ped_IVA, Ped_Transporte, Ped_Estado, Ped_Cancelado,
Ped_Recibe, Ped_Tipo from Pedidos, Clientes where Ped_Codigo=" & nPed_Codigo & " and
Pedidos.Cli_Codigo=" & nCli_Codigo & " and Clientes.Cli_Codigo=Pedidos.Cli_Codigo"
        set RSPedido=DBConn.Execute (strConsulta)
        if RSPedido.eof then
          Response.Write ("<br>")

```

```

        Response.Write ("<center><font face=verdana size=3><b>Los Datos
no son Correspondientes.</b></font></center>")
        Response.Write ("<br>")
        Response.Write ("<center><font face=verdana size=2><b><a
href=Ver_Pedidos.htm>Continuar</a></b></font></center>")
    else
        %>
            <center>
            <table>
                <tr>
                    <td>
                        <table>
                            <tr>
                                <td align=Right><font face=verdana size=2>Número de Cliente:</font></td>
                                <td align=Left><font
face=verdana size=2><%=RSPedido("Cli_Codigo")%></font></td>
                            </tr>
                            <tr>
                                <td align=Right><font face=verdana size=2>Nombre de Cliente:</font></td>
                                <td align=Left><font
face=verdana size=2><%=RSPedido("Cli_Nombres") & " " &
RSPedido("Cli_Apellidos")%></font></td>
                            </tr>
                        </table>
                    </td>
                    <td>
                        <table>
                            <tr>
                                <td align=Right><font face=verdana size=2>Número de Pedido:</font></td>
                                <td align=Left><font
face=verdana size=2><%=RSPedido("Ped_Codigo")%></font></td>
                            </tr>
                            <tr>
                                <td align=Right><font face=verdana size=2>Fecha de Pedido:</font></td>
                                <td align=Left><font
face=verdana size=2><%=RSPedido("Ped_Fecha")%></font></td>
                            </tr>
                        </table>
                    </td>
                </tr>
            </table>
            </center>
        <%
            strConsultaDet="Select Pedidos_Detalle.Art_Codigo, Art_Nombre,
Ped_Det_Cantidad, Ped_Det_Precio from Pedidos_Detalle, Articulos where Ped_Codigo=" &
RSPedido("Ped_Codigo") & " and Articulos.Art_Codigo=Pedidos_Detalle.Art_Codigo"
            set RSPedidoDet=DBConn.Execute (strConsultaDet)
        %>

```

```

set RSCliente=DBConn.Execute ("Select Cli_Apellidos_Recibe,
Cli_Nombres_Recibe from Clientes where Cli_Codigo=" & nCli_Codigo)
%>

```

```

<center>
<table border=1>
<tr>
<td align=Center><font face=verdana
size=1><b>Código</b></font></td>
<td align=Center><font face=verdana
size=1><b>Nombre</b></font></td>
<td align=Center><font face=verdana
size=1><b>Cantidad</b></font></td>
<td align=Center><font face=verdana
size=1><b>Precio Un.</b></font></td>
<td align=Center><font face=verdana
size=1><b>Total</b></font></td>
</tr>
<%

```

```

do While not RSPedidoDet.eof

```

```

nPrecioUn=round(RSPedidoDet("Ped_Det_Precio")/1.14,2)
Response.Write ("<tr>" & vbCrLf)
Response.Write ("<td
align=Right><font face=verdana size=2>" & RSPedidoDet("Art_Codigo") & "</font></td>" &
vbCrLf)
Response.Write ("<td
align=Left><font face=verdana size=2>" & RSPedidoDet("Art_Nombre") & "</font></td>" &
vbCrLf)
Response.Write ("<td
align=Right><font face=verdana size=2>" & FormatMoney(RSPedidoDet("Ped_Det_Cantidad")) &
"</font></td>" & vbCrLf)
Response.Write ("<td
align=Right><font face=verdana size=2>" & FormatMoney(nPrecioUn) & "</font></td>" & vbCrLf)
Response.Write ("<td
align=Right><font face=verdana size=2>" & FormatMoney(RSPedidoDet("Ped_Det_Cantidad")*
nPrecioUn) & "</font></td>" & vbCrLf)
nSubTotal = nSubTotal +
RSPedidoDet("Ped_Det_Cantidad")* nPrecioUn
Response.Write ("</tr>" & vbCrLf)
RSPedidoDet.MoveNext
loop
%>
<tr>
<td></td>
<td></td>
<td></td>
<td align=Right><font face=verdana
size=2>SubTotal:</font></td>
<td align=Right><font face=verdana
size=2><%=FormatMoney(nSubTotal)%></font></td>
</tr>
<tr>
<td></td>

```

```

                <td></td>
                <td></td>
                <td align=Right><font face=verdana
size=2>IVA:</font></td>
                <td align=Right><font face=verdana
size=2><%=FormatMoney(RSPedido("Ped_IVA"))%></font></td>
            </tr>
            <tr>
                <td></td>
                <td></td>
                <td></td>
                <td align=Right><font face=verdana
size=2>Transporte:</font></td>
                <td align=Right><font face=verdana
size=2><%=FormatMoney(RSPedido("Ped_Transporte"))%></font></td>
            </tr>
            <tr>
                <td></td>
                <td></td>
                <td></td>
                <td align=Right><font face=verdana
size=2>Total:</font></td>
                <td align=Right><font face=verdana
size=2><%=FormatMoney(nSubTotal+RSPedido("Ped_Transporte")+RSPedido("Ped_IVA"))%></fo
nt></td>
            </tr>
        </table>
    </center>
    <br>
    <form id=FrmAct_Pedido name=FrmAct_Pedido
action=Actualiza_Pedido.asp>
    <center>
        <table>
            <tr>
                <td><font face=verdana
size=2>Estado:</font></td>
                <td><font face=verdana
size=2>Cancelado:</font></td>
            </tr>
            <tr>
                <td>
                    <%
                        if
RSPedido("Ped_Estado")="Generado" then
                            Response.Write ("<input type=Radio Value=Generado name=Ped_Estado
Checked=TRUE><font face=verdana size=2>Por Entregar</font><br>")
                            Response.Write("<input type=Radio Value=Entregado name=Ped_Estado
disabled=false><font face=verdana size=2>Entregado</font><br>")
                        else

```

```

Response.Write("<input type=Radio Value=Generado name=Ped_Estado
disabled=false><font face=verdana size=2>Por Entregar</font><br>")

```

```

Response.Write("<input type=Radio Value=Entregado name=Ped_Estado
Checked=TRUE><font face=verdana size=2>Entregado</font><br>")

```

```

end if

```

```

%>

```

```

</td>

```

```

<td>

```

```

<%if

```

```

RSPedido("Ped_Cancelado") then

```

```

Response.Write

```

```

("<input Type=Radio name=Ped_Cancelado Value=True Checked=True><font face=verdana
size=2>Si</font><br>")

```

```

Response.Write

```

```

("<input Type=Radio name=Ped_Cancelado Value=False><font face=verdana size=2>No</font>")

```

```

else

```

```

Response.Write

```

```

("<input Type=Radio name=Ped_Cancelado Value=True><font face=verdana
size=2>Si</font><br>")

```

```

Response.Write

```

```

("<input Type=Radio name=Ped_Cancelado Value=False Checked=True><font face=verdana
size=2>No</font>")

```

```

end if%>

```

```

</td>

```

```

</tr>

```

```

</table>

```

```

<br>

```

```

<table>

```

```

<tr>

```

```

<td><font face=verdana size=2>Tipo

```

```

de Entrega</font></td>

```

```

</tr>

```

```

<tr>

```

```

<td>

```

```

<%

```

```

strRecibe=RSCliente("Cli_Nombres_Recibe") & " " & RSCliente("Cli_Apellidos_Recibe")

```

```

select case

```

```

RSPedido("Ped_Tipo")

```

```

case

```

```

"Almacen"

```

```

Response.Write("<input type=Radio Value=Almacen name=Ped_Tipo
Checked=TRUE><font face=verdana size=2>Retirar en un Almacén de Comercial
Puruhá</font><br>")

```

```

Response.Write("&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<font face=verdana
size=2>Recibe: </font><input name=Ped_Recibe type=Text Value="" & RSPedido("Ped_Recibe") &
""><font face=verdana size=2></font><br>")

```



```

        end if
    end if
%>
</form>
</body>
</html>

```

Archivo: Productos..asp

```

<% @ Language=VBScript %>
<% Option Explicit %>
<HTML>
<HEAD>
<META>
</HEAD>
<SCRIPT LANGUAGE=javascript RUNAT="Server">
    var WGdc=","; var WGgc=" "; var WGnc="("; var WGcs="";

    function FormatMoney(A,W) {
        var N=Math.abs(Math.round(A*100));
        var S=((N<10)?"00":((N<100)?"0":""))+N;
        S=WGcs+((A<0)?WGnc:"")+WGgroup(S.substring(0,(S.length-
2)))+WGdc+S.substring((S.length-2),S.length)+((A<0&&WGnc=="(")?")":"");
        return (S.length>W)?"Over":S;
    }

    function WGgroup(S) {
        return (S.length<4)?S:(WGgroup(S.substring(0,S.length-
3))+WGgc+S.substring(S.length-3,S.length));
    }
</SCRIPT>
<BODY bgColor=#ffffff leftMargin=40 link=#500000 vLink=#505050>
<FONT face="Verdana, Arial, Helvetica" size=2>
<TABLE border=2>
    <tr>
        <td><b>Código</b></td>
        <td><b>Artículo</b></td>
        <td align=right><b>Precio</b></td>
    </tr>
</%
Dim DBCon,RSArticulo
Dim strCon_Articulo
set DBCon= Server.CreateObject ("ADODB.Connection")
DBCon.Open "ConPuruha"
'set RSArticulo=server.CreateObject ("ADODB.Recordset")
strCon_Articulo="Select Art_Codigo, Art_Nombre, Art_Precio_Un From Articulos Where
Gru_Codigo="" & Request("cboProduct") & "" Order by Art_Nombre"
set RSArticulo=DBCon.Execute (strCon_Articulo)
do while not RSArticulo.EOF
    Response.Write ("<tr>" & vbCrLf)

```

```

        Response.Write ("<td>" & RSArticulo("Art_Codigo") & "</td>" & vbCrLf)
        Response.Write ("<td><a href=addcart.asp?Art_Cod=" & RSArticulo("Art_Codigo") & ">" &
RSArticulo("Art_Nombre") & "</a></td>" & vbCrLf)
        Response.Write ("<td align=right>" & FormatMoney(RSArticulo("Art_Precio_Un"),12) &
"</td>" & vbCrLf)
        Response.Write ("</tr>" & vbCrLf)
        RSArticulo.MoveNext
    loop
    RSArticulo.Close
    set RSArticulo=Nothing
    DBCon.Close
    set DBCon=Nothing
    '='<A HREF=AddCart.asp?Art_Cod=" + [Art_Codigo] + ">" + [Art_Nombre] + "</A>"
    %>
</TABLE>
</font>
</BODY>
</HTML>

```

Archivo: Viewcart.asp

```

<% @ Language=VBScript %>
<% Option Explicit %>
<html>
<head></head>
<body bgColor="#ffffff" leftMargin="40" link="#500000" vLink="#505050">

    <script LANGUAGE="javascript" RUNAT="Server">

        var WGdc=","; var WGgc="."; var WGnc="("; var WGcs=" ";

        function FormatMoney(A,W) {
            var N=Math.abs(Math.round(A*100));
            var S=((N<10)?"00":((N<100)?"0":""))+N;
            S=WGcs+((A<0)?WGnc:"")+WGgroup(S.substring(0,(S.length-
2)))+WGdc+S.substring((S.length-2),S.length)+((A<0&&WGnc=="(")?"):"");
            return (S.length>W)?"Over":S;
        }

        function WGgroup(S) {
            return (S.length<4)?S:(WGgroup(S.substring(0,S.length-
3))+WGgc+S.substring(S.length-3,S.length));
        }

    </script>
    <table border="2">
        <tr>
            <td><font face="Verdana" size="1"><b>Código</b></font></td>
            <td><font face="Verdana" size="1"><b>Nombre</b></font></td>
            <td><font face="Verdana" size="1"><b>Cant.</b></font></td>
            <td><font face="Verdana" size="1"><b>Pre. Uni.</b></font></td>
            <td><font face="Verdana" size="1"><b>Total</b></font></td>

```

```

        <td><font face="Verdana" size="1"><b>Mod./Quitar</b></font></td>
    </tr>
<%
    Dim DBConn, RSDetallePedido
    Dim strConsulta, strArt_Codigo
    Dim nSubTotal, nAux, nPrecioUn
    set DBConn=server.CreateObject ("ADODB.Connection")
    DBConn.Open "ConPuruha"
    set RSDetallePedido=server.CreateObject ("ADODB.Recordset")
    strConsulta="Select Pendiente_Pedido.Art_Codigo, Art_Nombre, Pen_Det_Cantidad,
    Pen_Det_Precio from Pendiente_Pedido, Articulos Where Pen_SessionID="" & Session.SessionID &
    "" and Articulos.Art_Codigo=Pendiente_Pedido.Art_Codigo"
    'Response.Write (strConsulta)
    set RSDetallePedido=DBConn.Execute (strConsulta)
    'strArt_Codigo=RSDetallePedido("Art_Codigo")
    nSubTotal=0
    do while not RSDetallePedido.EOF
        Response.Write ("<tr>" & vbCrLf)
        Response.Write ("<td><FONT face=Verdana size=1>" &
    RSDetallePedido("Art_Codigo") & "</font></td>" & vbCrLf)
        Response.Write ("<td><FONT face=Verdana size=1>" &
    RSDetallePedido("Art_Nombre") & "</font></td>" & vbCrLf)
        Response.Write ("<td align=Right><FONT face=Verdana size=1>" &
    RSDetallePedido("Pen_Det_Cantidad") & "</font></td>" & vbCrLf)
        nPrecioUn=round (RSDetallePedido("Pen_Det_Precio")/1.12,2)
        Response.Write ("<td align=Right><FONT face=Verdana size=1>" &
    FormatMoney(nPrecioUn) & "</font></td>" & vbCrLf)
        Response.Write ("<td align=Right><FONT face=Verdana size=1>" &
    FormatMoney(nPrecioUn * RSDetallePedido("Pen_Det_Cantidad")) & "</font></td>" & vbCrLf)
        Response.Write ("<td><FONT face=Verdana size=1><A
    HREF=SCUpCan.asp?Art_Cod="" & RSDetallePedido("Art_Codigo") & ">Modificar</A> / <A
    HREF=SCDelete.asp?Art_Cod="" & RSDetallePedido("Art_Codigo") & ">Quitar</A>" &
    "</font></td>" & vbCrLf)
        Response.Write ("</tr>" & vbCrLf)
        nSubTotal=nSubTotal + nPrecioUn * RSDetallePedido("Pen_Det_Cantidad")
        RSDetallePedido.MoveNext
    loop

    'Response.Write ("</table>" & vbCrLf)
    nAux=nSubTotal
    Response.Write ("<tr>" & vbCrLf)
    Response.Write ("<td>" & "" & "</td>" & vbCrLf)
    Response.Write ("<td>" & "" & "</td>" & vbCrLf)
    Response.Write ("<td>" & "" & "</td>" & vbCrLf)

    Response.Write ("<td><Table>" & vbCrLf)

        Response.Write ("<tr>" & vbCrLf)
        Response.Write ("<td align=Right><FONT face=Verdana size=1>" &
    "SubTotal" & "</font></td>" & vbCrLf)
        Response.Write ("</tr>" & vbCrLf)

    Response.Write ("<tr>" & vbCrLf)

```

```

                Response.Write("<td align=Right><FONT face=Verdana size=1>" &
"IVA" & "</font></td>" & vbCrLf)
                Response.Write("</tr>" & vbCrLf)

                Response.Write("<tr>" & vbCrLf)
                Response.Write("<td align=Right><FONT face=Verdana size=1>" &
"Transporte" & "</font></td>" & vbCrLf)
                Response.Write("</tr>" & vbCrLf)

                Response.Write("<tr>" & vbCrLf)
                Response.Write("<td align=Right><FONT face=Verdana size=1>" &
"Total" & "</font></td>" & vbCrLf)
                Response.Write("</tr>" & vbCrLf)

                Response.Write("</table></td>" & vbCrLf)

                Response.Write("<td><Table>" & vbCrLf)

                Response.Write("<tr>" & vbCrLf)
                Response.Write("<td align=Right><FONT face=Verdana size=1>" &
FormatMoney(nAux,12) & "</font></td>" & vbCrLf)
                Response.Write("</tr>" & vbCrLf)

                Response.Write("<tr>" & vbCrLf)
                Response.Write("<td align=Right><FONT face=Verdana size=1>" &
FormatMoney(nAux*0.12,12) & "</font></td>" & vbCrLf)
                Response.Write("</tr>" & vbCrLf)

                Response.Write("<tr>" & vbCrLf)
                Response.Write("<td align=Right><FONT face=Verdana size=1>" &
FormatMoney(0,12) & "</font></td>" & vbCrLf)
                Response.Write("</tr>" & vbCrLf)

                Response.Write("<tr>" & vbCrLf)
                Response.Write("<td align=Right><FONT face=Verdana size=1>" &
FormatMoney(nAux*1.12,12) & "</font></td>" & vbCrLf)
                Response.Write("</tr>" & vbCrLf)

                Response.Write("</table></td>" & vbCrLf)

                Response.Write("</tr>" & vbCrLf)
                Response.Write("</table>" & vbCrLf)

                if RSDetallePedido.RecordCount =0 then%>
                    <font face="Verdana"><font face="verdana" size="1">No hay productos en la
compra.</font>
                    <%else%>
                        <font face="verdana" size="2"><a href="SCClear.asp">Borrar contenido del
carrito</a> <a href=" ../Shopcart/TipoEntrega.htm">Guardar pedido</a></font>
                    <%end if
                Session("SubTotal")=nSubTotal

                RSDetallePedido.Close

```

```

        set RSDetallePedido=Nothing
        DBConn.Close
        set DBConn=Nothing

%>

</font></font></table>
</body>
</html>

Archivo: SaveOrder.asp
<% @ Language=VBScript %>
<% Option explicit %>
<HTML>
<HEAD>
<META>
</HEAD>
<BODY onload=printPage()>
    <SCRIPT LANGUAGE=javascript Runat="Server">
        var WGdc=","; var WGgc="."; var WGnc="("; var WGcs="";
        function FormatMoney(A,W) {
            var N=Math.abs(Math.round(A*100));
            var S=((N<10)?"00":((N<100)?"0":""))+N;
            S=WGcs+((A<0)?WGnc:"")+WGgroup(S.substring(0,(S.length-
2)))+WGdc+S.substring((S.length-2),S.length)+((A<0&&WGnc=="(")?"):"");
            return (S.length>W)?"Over":S;
        }
        function WGgroup(S) {
            return (S.length<4)?S:(WGgroup(S.substring(0,S.length-
3)))+WGgc+S.substring(S.length-3,S.length);
        }
    </SCRIPT>
<%
    Dim strIngresarPedido, strIngresarDetalle, strConsultaPendiente
    Dim strDestinatario, strAsunto, strCuerpo
    Dim nPed_Codigo
    dim DBConn, oMail, RSPedidos, RSPendientes, RSCliente
    Dim strPed_Recibe, strPed_Tipo
    Dim nPed_Transporte, nIVA

    Response.Write ("<center><font face=verdana size=3><b>Inicio : " & Timer &
"</b></font></center>")

    set DBConn=Server.CreateObject ("ADODB.Connection")
    DBConn.Open "ConPuruha"
        set RSPedidos=DBConn.Execute ("select Ped_Codigo from Pedidos order by
Ped_Codigo DESC")
        nPed_Codigo=RSPedidos("Ped_Codigo")
        nPed_Codigo=nPed_Codigo*1
        nPed_Codigo=nPed_Codigo+1
    Session("Pedido")=nPed_Codigo
    'Session("Mails")=Session("Mails") & vbCrLf
    'Session("Mails")=Session("Mails") & "Datos del Pedido" & vbCrLf

```

```

strPed_Tipo=Request.Form("Ped_Entrega")
select case strPed_Tipo
case "Almacen" :
                                strPed_Recibe=Request.Form("TxtRecibe")
                                nPed_Transporte=0

case "Personal":
                                strPed_Recibe=""
                                nPed_Transporte=0

case "Obra":
                                strPed_Recibe=""
                                nPed_Transporte=Session("SubTotal")*1.14*0.04

end select
nIVA=FormatMoney((Session("SubTotal"))*0.14,12)
strIngresarPedido="Insert into Pedidos Values(" & nPed_Codigo & "," &
Session("ClienteID") & "," & Date & "," & FormatMoney(Session("SubTotal"),12) & "," & nIVA
& "," & FormatMoney(nPed_Transporte,12) & "," & 'Generado' & " & "False" & "," & strPed_Recibe &
"," & strPed_Tipo & ")"
'***** ojo aqui debo consultar el último pedido aumentar uno para grabar.

'Response.Write (strIngresarPedido)
DBConn.Execute (strIngresarPedido)
'Aqui empiezo transacción
strConsultaPendiente="select * from Pendiente_Pedido where Pen_SessionID=" &
Session.SessionID & ""
set RSPendientes=DBConn.Execute (strConsultaPendiente)
do while not RSPendientes.EOF
    strIngresarDetalle="Insert into Pedidos_Detalle Values (" & nPed_Codigo & "," &
RSPendientes("Art_Codigo") & "," & RSPendientes("Pen_Det_Cantidad") & "," &
RSPendientes("Pen_Det_Precio") & ")"
    DBConn.Execute (strIngresarDetalle)
    RSPendientes.MoveNext

loop
'Aqui Termino Transacción
'Session("Mails")=Session("Mails") & "Detalle" & vbCrLf & Session("DetalleTemp") &
vbCrLf & "Fin del Pedido" & vbCrLf
DBConn.Execute ("Delete * from Pendiente_Pedido where Pen_SessionID=" &
Session.SessionID & """)
'Datos del Cliente
set RSCliente=DBConn.Execute ("Select Cli_Apellidos, Cli_Nombres, Cli_Cedula,
Cli_Direccion_Exter, Cli_Ciudad_Exter, Cli_Pais From Clientes Where Cli_Codigo=" &
Session("ClienteID") )
%>
<SCRIPT language=JavaScript><!--
<!-- The JavaScript Source!! http://javascript.internet.com -->

<!--Begin
function printPage()
{
    if (window.print){
        agree = confirm('Esta es una solicitud lista para imprimir.
\nSi preciona Cancel Imprima después desde el menú (Archivo) \n\nPresione OK para imprimir');
        if (agree) window.print();

```

```

}}
// -->
</SCRIPT>
<form name=FrmGiro>
  <p align=right><font face=verdana size=3>Valor $:
<b><%=FormatMoney(Session("SubTotal") + nIVA + nPed_Transporte,12)%></b></font></p>
  <p align=right><font face=verdana size=1>Referencia No. de Pedido
<b><%=nPed_Codigo%></b></font></p>
  <p align=left><font face=verdana size=3><B><U>DATOS DEL
CLIENTE</U></B></font></p>
  <table>
    <tr>
      <td align=left><font face=verdana size=2>NOMBRE:</font></td>
      <td align=left><font face=verdana
size=2><b><%=RSCliente("Cli_Apellidos") & " " & RSCliente("Cli_Nombres")%></b></font></td>
    </tr>
    <tr>
      <td align=left><font face=verdana size=2>CEDULA:</font></td>
      <td align=left><font face=verdana
size=2><b><%=RSCliente("Cli_Cedula")%></b></font></td>
    </tr>
    <tr>
      <td align=left><font face=verdana
size=2>DIRECCION:</font></td>
      <td align=left><font face=verdana
size=2><b><%=RSCliente("Cli_Direccion_Exter")%></b></font></td>
    </tr>
    <tr>
      <td align=left><font face=verdana size=2>CIUDAD:</font></td>
      <td align=left><font face=verdana
size=2><b><%=RSCliente("Cli_Ciudad_Exter")%></b></font></td>
    </tr>
    <tr>
      <td align=left><font face=verdana size=2>PAIS:</font></td>
      <td align=left><font face=verdana
size=2><b><%=RSCliente("Cli_Pais")%></b></font></td>
    </tr>
  </table>

  <p align=left><font face=verdana size=3><B><U>DATOS DEL
BENEFICIARIO</U></B></font></p>
  <table>
    <tr>
      <td align=left><font face=verdana size=2>NOMBRE:</font></td>
      <td align=left><font face=verdana size=2><b>COMPUTEC CIA.
LTDA.</b></font></td>
    </tr>
    <tr>
      <td align=left><font face=verdana size=2>CUENTA
No.:</font></td>
      <td align=left><font face=verdana
size=2><b>15249618</b></font></td>
    </tr>
  </table>

```

	<td align="left">DIRECCION:</td>	DIRECCION:	
	<td align="left">AV. MIGUEL ANGEL LEON 32-28</td>	AV. MIGUEL ANGEL LEON 32-28	
	<td align="left">CIUDAD:</td> <td align="left">RIOBAMBA</td>	CIUDAD:	RIOBAMBA
	<td align="left">PAIS:</td> <td align="left">ECUADOR</td>	PAIS:	ECUADOR

DATOS DEL BANCO DE DESTINO

	<td align="left">NOMBRE:</td> <td align="left">BANCO DEL PICHINCHA</td>	NOMBRE:	BANCO DEL PICHINCHA
	<td align="left">ABA:</td> <td align="left">066012867</td>	ABA:	066012867
	<td align="left">SWIFT:</td> <td align="left">PICHECEQ</td>	SWIFT:	PICHECEQ
	<td align="left">DIRECCION:</td> <td align="left">1° CONSTITUYENTE Y GARCIA MORENO</td>	DIRECCION:	1° CONSTITUYENTE Y GARCIA MORENO
	<td align="left">CIUDAD:</td> <td align="left">RIOBAMBA</td>	CIUDAD:	RIOBAMBA
	<td align="left">PAIS:</td> <td align="left">ECUADOR</td>	PAIS:	ECUADOR


```

        <br>
        <br>
        <center>
        <table>
            <tr>
                <td><font face=verdana
size=2>NOMBRE:_____</font></td>
                <td><font face=verdana
size=2>FIRMA:_____</font></td>
            </tr>
            <tr>
                <td><font face=verdana
size=2>CI/RUC:_____</font></td>
            </tr>
        </table>
        </center>
    </form>
    <%
        Response.Write ("<center><font face=verdana size=3><b>Final : " & Timer &
"</b></font></center>")
        DBConn.Close
        RSClient.Close
        set DBConn=Nothing
        set RSClient=Nothing
        set oMail=Nothing
    %>
</BODY>
</HTML>

```

ANEXO IV

CONFIGURACIONES AVANZADAS DE LA CONEXIÓN A BASE DE DATOS

CAMPOS OPCIONALES DE LOS ARCHIVOS .IDC

Para poder explicar cada uno de estos parámetros utilizaremos como referencia a Microsoft Windows NT Server que utiliza ODBC a través del servicio de Microsoft Internet Information Server (IIS Services) para SQLServer, el cual se tomará como referencia para ejemplificar nuestro análisis.

DefaultParameters = parám = valor [, parám = valor] [...].- Los valores de los parámetros, si hay alguno, que se van a utilizar en el archivo del Conector de Bases de datos de Internet siempre que el cliente no especifique algún parámetro.

Expires.- El número de segundos que hay que esperar antes de actualizar una página con salida en caché. Si una petición posterior es idéntica, la página de caché se devolverá sin tener acceso a la Base de datos. De forma predeterminada, el IDC no pone en caché las páginas de salida. Sólo las incluye en caché cuando se utiliza el campo Expires *Este campo es útil cuando desee forzar una nueva consulta de la Base de datos tras un determinado periodo de tiempo.*

MaxFieldSize.- El espacio de búfer máximo por campo que asigna el IDC. Los caracteres posteriores a éste se truncarán. El parámetro se aplica sólo a aquellos campos de la Base de datos que superen el valor de este parámetro (Ej. 8192 bytes). *Este parámetro es útil cuando se trabaja con campos (atributos) cuyo dominio es inferior (menor al 70%) de la capacidad por defecto del parámetro, evitando que se desperdicie por reserva espacio innecesario en el búfer.*

MaxRecords.- El número máximo de registros que el IDC devolverá de cualquier consulta. El valor MaxRecords no se define de forma predeterminada, lo que significa que una consulta puede devolver hasta 4.000 millones de registros. *Es recomendable definir este valor para limitar los registros devueltos y evitar un bottle neck en el IDC o el uso indiscriminado de recursos (memoria, procesador, etc).*

ODBCConnection.- Es recomendable insertar este campo con el valor de *agrupación* para agregar la conexión a la agrupación de conexiones, que conserva la conexión con la Base de datos abierta para futuras peticiones, por lo que el IDC enviará los datos a través de una conexión agrupada para la posterior ejecución de un archivo .idc que contiene los mismos valores de Datasource, Username y Password. *Esta opción es útil para mejorar el rendimiento utilizando el Conector de Bases de datos de Internet.*

Adicionalmente, hay una opción *sin agrupación*, que especifica que la conexión del archivo .idc en la que se define esta opción no debe tomarse de la agrupación de conexiones, ya que si hay un límite en cuanto al número de conexiones actuales, no es dable que la agrupación de conexiones monopolice todas las conexiones; de lo contrario, nadie podría conectarse a SQL Server, debiéndose para ello definir el valor de este campo como *nopool* *Esto permite administrar la caché de conexiones con más precisión.*

Passwords.- La contraseña que corresponde al nombre de usuario. Si no hay ninguna contraseña, este campo puede dejarse en blanco. *Es recomendable que se defina una contraseña para cada grupo de usuarios, incluso para aquellos que ingresen como anónimo se podría predefinir este campo como anonymous, de tal manera que se pueda establecer privilegios en cuanto a ejecución de peticiones.*

RequiredParameters.- Establece los nombres de los parámetros, si hay alguno, que se asegura que pasarán del cliente; de lo contrario, devolverá un error. Los nombres de los parámetros se separan mediante comas. *Este parámetro es útil para configurar la petición SQL.*

Translationfile.- Indica la ruta de acceso al archivo que asigna caracteres no ingleses (como à, ô o é) para que los exploradores puedan mostrarlos adecuadamente en formato HTML. Si el archivo de traducción no está en el mismo directorio que el archivo .idc, debe escribir la ruta de acceso completa al archivo de traducción. La sintaxis es:

translationfile: C:\nombre_directorio\nombre_archivo

En este sentido, si se publica una Base de datos en un idioma que no sea inglés, el campo Translationfile, indicará su archivo de traducción, en el que los caracteres especiales se asignan en el siguiente formato: valor=cadena<CR>, donde valor es un carácter internacional y cadena es el código de traducción de HTML.

Username.- Indica un nombre de usuario válido para el origen de datos proporcionado por el campo Datasource.

Content-Type.- Especifica cualquier tipo MIME válido que describa qué se devolverá al cliente. Si el archivo .htx contiene HTML, será "text/html".

CAMPOS OPCIONALES AVANZADOS DE ODBC

Las opciones avanzadas de ODBC permiten configurar el controlador ODBC utilizado por el Conector de Bases de datos de Internet para depurarlo y ajustarlo a las características del hardware y software utilizado mejorando la interacción entre sus componentes. El formato del archivo IDC es:

ODBCOptions: Nombre_opción=Valor[,Nombre_opción=Valor...]

Por ejemplo, para que la instrucción SQL deje de ejecutarse durante más de 10 segundos y activar el seguimiento de las llamadas a funciones ODBC en el archivo IDC tiene que especificar lo siguiente:

```
ODBCOptions: SQL_QUERY_TIMEOUT=10,  
SQL_OPT_TRACE=1,  
SQL_OPT_TRACEFILE=C:\Sql.log
```

A continuación estudiamos todas las opciones que permiten manejar estas configuraciones. El formato a seguir para nuestra explicación es el siguiente:

Nombre de la opción: Dominio de valores
Propósito

SQL_ACCESS_MODE: 0 = Lectura/escritura, 1 = Sólo lectura

Es el indicador para el controlador de ODBC u origen de datos de que no requiere la conexión para ser compatible con instrucciones SQL que producen actualizaciones. *Este modo puede utilizarse para optimizar estrategias de bloqueo, la administración de las transacciones u otras áreas del controlador o el origen de datos.*

El comportamiento del controlador y del origen de datos cuando se les pide que procesen instrucciones SQL que no son de sólo lectura durante una conexión de sólo lectura está definido por la implementación. De forma predeterminada SQL_ACCESS_MODE se establece como 0, lo que permite la lectura y la escritura.

SQL_LOGIN_TIMEOUT: Entero

Es el número de segundos que hay que esperar para que finalice un inicio de sesión antes de desconectarse. El valor predeterminado depende del controlador y debe ser distinto de cero. Si el valor es 0, el tiempo de espera se inhabilita y un intento de conexión esperará indefinidamente. Si el tiempo

de espera especificado sobrepasa el tiempo de espera de inicio de sesión máximo del origen de datos, el controlador sustituirá dicho valor. *Es recomendable que estadísticamente se estime un valor aproximado para este parámetro de tal manera que el intento de conexión no permanezca indefinidamente.*

SQL_OPT_TRACE: 0 = Desactivar el seguimiento, 1 = Activar el seguimiento

Cuando se activa el seguimiento, todas las llamadas a funciones ODBC que se realiza se escriben en el archivo de seguimiento. Con la opción SQL_OPT_TRACEFILE se puede especificar un archivo de seguimiento. Si este archivo ya existe, el controlador ODBC se agrega al archivo. De lo contrario, crea el archivo. Si se activa el seguimiento sin especificar ningún archivo de seguimiento, ODBC escribe en el archivo Sql.log. *El contenido del archivo .log proporciona información útil para poder realizar posteriores y continuos afinamientos.*

SQL_OPT_TRACEFILE: Nombre de archivo

El nombre del archivo de seguimiento que se va a utilizar cuando SQL_OPT_TRACE=1. El valor predeterminado es SQL.LOG. *Es recomendable guardar en dispositivos externos una bitácora de los mismos para posteriores análisis de comportamiento del sistema, evitando que ocupen espacio de almacenamiento que podría utilizarse. Además que se evitaría que los .log sean muy grandes con las repercusiones que ello implica.*

SQL_PACKET_SIZE: Entero

Indica el tamaño del paquete de red, en bytes, que va a utilizarse para intercambiar información entre el sistema de la administración de Bases de datos (DBMS) y el servidor Web. Sin embargo, muchos orígenes de datos no son compatibles con esta opción o sólo pueden devolver el tamaño del paquete de red. *Si el tamaño especificado sobrepasa o es menor que el tamaño mínimo del paquete, el controlador sustituye dicho valor. Si el origen de datos lo permite es recomendable sincronizar este el tamaño de paquete de red.*

SQL_TRANSLATE_DLL: Nombre de archivo

El nombre de una DLL que contiene las funciones `SQLDriverToDataSource` y `SQLDataSourceToDriver`, y que el controlador carga y utiliza para realizar tareas como la traducción del juego de caracteres. Será útil y necesario establecer este archivo en el caso de que se maneje software que requiere compatibilidad que no maneja la misma sintaxis que el controlador de origen.

SQL_TRANSLATE_OPTION: Entero

Este valor controla la funcionalidad de traducción, que es específica de la DLL de traducción utilizada.

SQL_TXN_ISOLATION: Entero. 1 = Lectura no confirmada, 2 = Lectura confirmada, 4 = Lectura repetible, 8 = Serializable, 16 = Control de versiones

Define el nivel de aislamiento de las transacciones. El Conector de Bases de datos de Internet no es compatible con las transacciones que se extienden más que la petición del archivo `.idc`. *Sin embargo, en el caso de algunos DBMS, definir la opción `SQL_TXN_ISOLATION` como 1 (Lectura no confirmada) dará como resultado una mayor concurrencia y, por tanto, un mejor rendimiento. No obstante, con esta configuración pueden recuperarse aquellos datos que otras transacciones no hayan confirmado con la Base de datos.*

SQL_MAX_LENGTH: Entero

Representa la cantidad máxima de datos de una columna de caracteres o binaria que devuelve el controlador. *Esta opción reduce el tráfico de red y sólo debe utilizarse cuando el origen de datos (lo opuesto al controlador) de un controlador con varias capas pueda implementarlo.*

SQL_MAX_ROWS: Entero

Indica el número máximo de filas que devuelve una instrucción SELECT. Si el valor es 0 (el valor predeterminado), el controlador devuelve todas las filas. *Esta opción reduce el tráfico de red cuando el propio origen de datos puede limitar las filas que se devuelven, frente a la variable incorporada MaxRecords del Conector de Bases de datos de Internet, que limita las filas recuperadas.*

SQL_NOSCAN: 0 = Buscar y convertir cláusulas de escape, 1 = No buscar y convertir cláusulas de escape

Especifica si no es necesario que el controlador busque cadenas SQL en cláusulas de escape. Si se define como 0 (el valor predeterminado), el controlador busca cláusulas de escape en cadenas SQL. Si se define como 1, el controlador no busca cláusulas de escape en cadenas SQL; en su lugar el controlador envía la instrucción directamente al origen de datos. *En caso de que la instrucción SQL no contenga ninguna cláusula de escape ODBC, la utilización de una sintaxis especial encerrada entre llaves ({ }) y la posterior definición de esta opción como 1 proporcionará una pequeña mejora en el rendimiento indicando al controlador que no debe buscar en la cadena SQL.*

SQL_QUERY_TIMEOUT: Entero. 0 = Sin tiempo de espera

Indica el tiempo en segundos que hay que esperar para que se ejecute una instrucción SQL antes de cancelar la consulta. Si se define como 0 (el valor predeterminado) no hay tiempo de espera. Si el tiempo de espera especificado sobrepasa el tiempo de espera máximo del origen de datos o es menor que el tiempo de espera mínimo, el controlador sustituirá dicho valor. *Es recomendable definir un valor de acuerdo al tiempo promedio de las transacciones mas frecuentes (utilizando la regla 80/20 mencionada en capítulos anteriores).*

Entero: Específico del controlador

Los valores de la opción específicos del controlador pueden especificarse con el formato *número=valor*. Por ejemplo, 4322 = 1, 234 = Cadena. Por lo general esta acepción se utiliza cuando la base de datos maneja tiene predefinido valores de uso exclusivo.