

INTRODUCCION

El presente trabajo es realizado con la finalidad de optar al título de ingeniero de sistemas, para ello se decidió realizar un aplicativo de orientación web el cual tiene por finalidad proveer una solución de comercio electrónico a la medida y al alcance económico de las pequeñas y medianas empresas de la ciudad de Bogotá que tengan por objeto social la comercialización de productos y/o servicios, dicha solución estará provista de opciones que le permitirán a la empresa que adquiera este aplicativo cargar los productos y clientes que tenga en su sistema actual a la base de datos del sitio web, esto haciendo uso del lenguaje xml.

Así como también permite realizar un mantenimiento rápido y de manera modularizada, esto debido a que este aplicativo esta desarrollado con base a la arquitectura propuesta por Microsoft llamada Windows DNA. La base de datos utilizada para este aplicativo es una de la base de datos mas segura y comercializada actualmente.

1. PLANTEAMIENTO DEL PROBLEMA

Debido al avance tecnológico, las compañías comercializadoras de productos y servicios, se ven sujetas a buscar nuevas tecnologías entre estas las negociaciones electrónicas, es decir que sus clientes desde cualquier lugar del mundo puedan acceder a la información de sus productos, ordenar pedidos y despachos en línea, pero como en el mercado esto implica altos costos de hosting, dominios, diseño y desarrollo, muchas de estas compañías se ven relegadas a seguir trabajando con el sistema tradicional, además uno de los puntos más neurálgicos en dichas negociaciones es el tener la información del inventario actualizada.

2. DESCRIPCION

En la actualidad todos los comerciantes de productos y servicios desean poner sus negocios en Internet, para ello se requiere adquirir una solución de comercio electrónico, pero los aplicativos de esta índole que actualmente existen son muy costosos o no cumplen con las expectativas del estilo de negociar en Colombia.

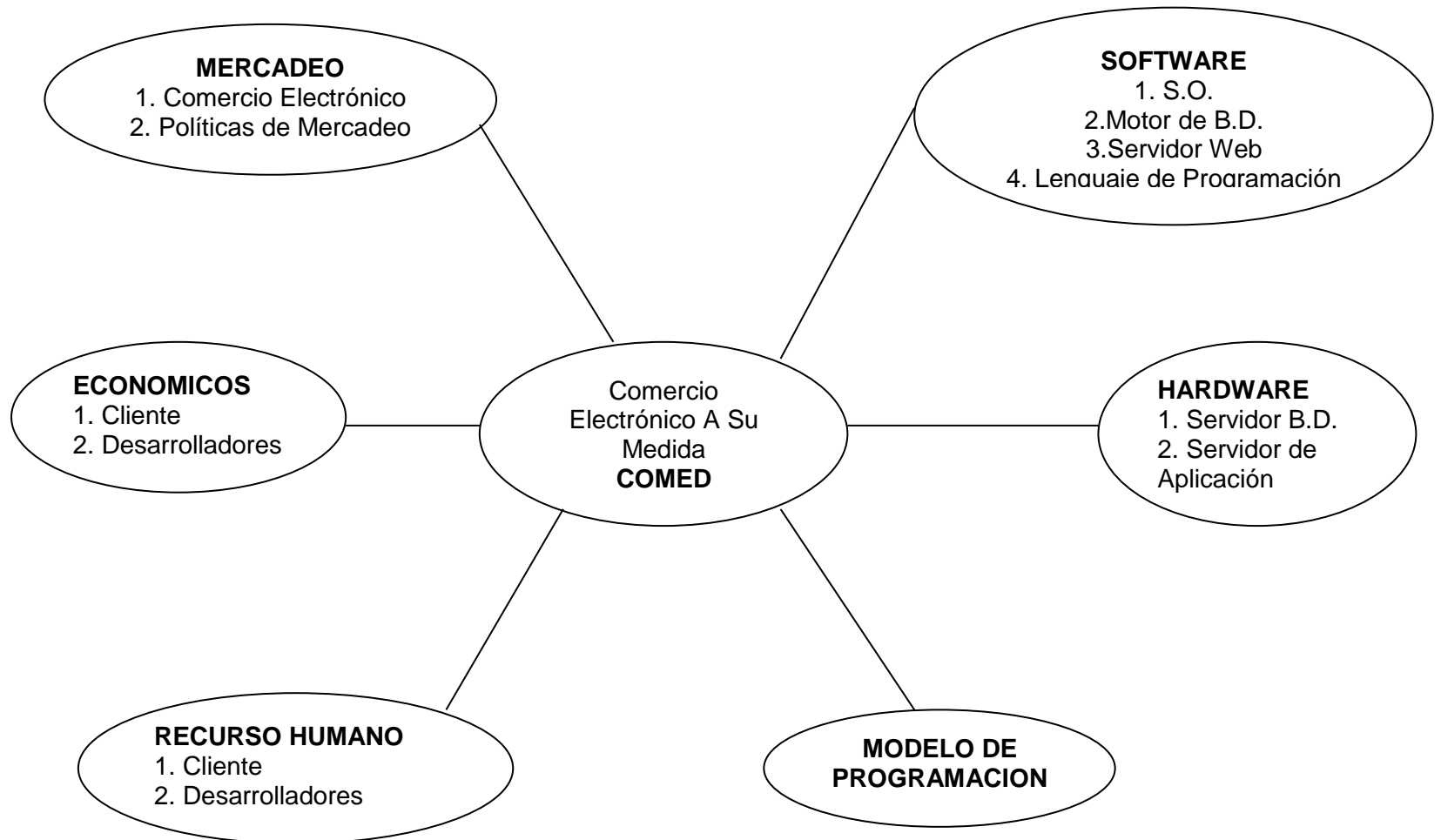
Además uno de los grandes inconvenientes que se presenta es el tener la información actualizada (clientes, productos, pedidos) en el sitio web, este es uno de los procesos más complicados y costosos al que se ven enfrentadas las empresas, debido a que las herramientas utilizadas para dicho propósito son de última tecnología y esto implica un gasto adicional en la capacitación de personal para el manejo de estas. Igualmente las soluciones de comercio electrónico del mercado actual están pensadas para dar su óptimo rendimiento en una estructura de sistemas distribuidos, es decir: un servidor de base de datos, un servidor Web y un servidor de aplicación, este tipo de infraestructura genera un mayor costo por el mantenimiento y seguridad que exige. Por lo anterior se decidió desarrollar una solución de comercio electrónico a la medida y al alcance de cualquier empresario o microempresario del país.

Para poder dar una garantía del producto final se adoptará el modelo de programación DNA propuesto por Microsoft, el cual divide la estructura de un aplicativo en tres partes conocido también con el nombre de tres capas; lo cual internamente maneja el concepto de sistemas distribuidos, la primera capa de abajo hacia arriba es la de la base de datos, en este nivel se ubican la base de datos de la aplicación como tal y de todos aquellos sistemas que interactúen de

una o de otra forma con la misma. En la segunda capa encontramos los componentes que contienen la lógica del negocio, es decir las políticas internas de la compañía. Por último tenemos la presentación de la información al usuario final, en este nivel se encuentran el servidor web, las páginas dinámicas(Asp, xml, dhtml, etc.) y las imágenes.

Uno de los aspectos más importantes a tener en cuenta para que este modelo de programación arroje los resultados esperados es tener un buen motor de base de datos, que para este caso será MySql el cual tiene un nivel seguridad muy alto, ya que permite la creación de diversos perfiles de usuarios para interactuar con la base de datos. Además para la transferencia de información se utilizará el lenguaje XML, el cual es muy sólido y confiable en este tipo de tareas, tanto así que la experiencia demuestra que posee un 95% de seguridad con relación a otros lenguajes o medios utilizados para tal fin.

3. MAPA DE VARIABLES



3.1 SOFTWARE

1. Sistema Operativo:

S.O.	COSTO	COMPONENTES	ASP	SEGURIDAD
LINUX	GRATUITO	NO	NO	CONFIABLE
WINDOWS	LICENCIA	SI	SI	POCO CONFIABLE

De acuerdo al paralelo anterior se aprecia notablemente que lo más aconsejable es adoptar un sistema operativo específico para cada una de las capas de la aplicación, ya que cada uno de estos ofrece beneficios que apuntan de manera directa a una u otra capa de la aplicación. Es decir la seguridad que ofrece Linux indica que tener la base de datos sobre él será muy seguro, por el contrario el manejo de componentes es más indicado sobre Windows.

2. Motor de base de datos:

MOTOR BD	SISTEMA OPERATIVO	LICENCIA GRATUITA
ORACLE	Linux, Windows, Unix	NO
SQL SERVER	Windows	NO
INFORMIX	Unix	NO
MYSQL	Linux, Unix, Windows	SI

Además de lo que se aprecia en el cuadro comparativo anterior, vale la pena decir que MySql es un motor de base de datos muy completo y esta desarrollado bajo licencia GLP lo cual significa que es de código abierto y de licencia gratuita, además de esto brinda un alto nivel de seguridad, por tanto para la elaboración del presente proyecto lo más aconsejable es usar MySql sobre Linux, ya que este es su sistema operativo nativo.

3. Servidor web:

COMED

SERVIDOR WEB	COSTO	COMPONENTES	ASP
APACHE	GRATUITO	NO	SI
IIS	S.O.	SI	SI

Como se observa en el cuadro anterior para el desarrollo de la aplicación bajo el modelo de programación Windows DNA lo más acertado es el uso del servidor web Internet Information Server.

4. Lenguaje a utilizar para el desarrollo de la aplicación:

LENGUAJE	MANEJA COMPONENTES	VALOR HOSTING
PHP	NO	GRATUITO HASTA \$12 US POR MES
ASP	SI	GRATUITO HASTA \$45 US POR MES
JSP	SI	HASTA \$85 US POR MES

Como se puede apreciar en el cuadro comparativo el lenguaje Asp (Active Server Pages) tiene manejo de componentes y además su costo de hosting no están alto. Por tanto aplica de manera directa sobre el modelo de programación de Microsoft DNA, el cual ha sido seleccionado para el desarrollo de este aplicativo.

3.2 HARDWARE

Es decir el equipo de cómputo que se requiere para la implementación del sitio, este se conseguirá a través de los proveedores de hosting existentes en el mercado, las especificaciones técnicas que deben cumplir estos son:

1. Un servidor para base de datos: Con sistema operativo Linux, MySql server mínimo versión 4.0.9 Gamma.

2. Un servidor para la aplicación: Con sistema operativo Windows 2000 server, Internet Information Server 5.0, servidor de componentes COM+.

3.3 MODELO DE PROGRAMACION

MODELO	MANTENIMIENTO	ESCALABILIDAD
TRADICIONAL	COMPLEJO	COMPLEJA
DNA	SENCILLO	SENCILLA

Al comparar los modelos de programación, se puede apreciar claramente que para el futuro de la aplicación y de quien la adquiera es mucho más rentable, que se desarrolle con el modelo de programación DNA propuesto por Microsoft, ya que en la forma en que este divide una aplicación da la posibilidad de un mantenimiento y escalabilidad más sencillo y por capa o parte.

3.4 RECURSO HUMANO

1. Cliente: este deberá designar a una persona para que se encargue de la administración de la aplicación, para esto dicha persona recibirá la correspondiente capacitación por parte del equipo de desarrollo de la misma.
2. Desarrolladores: Para el desarrollo de este aplicativo será necesario la intervención de un Ingeniero de Sistemas en el rol de asesor y dos tecnólogos en el rol de analistas y desarrolladores del mismo.

3.5 ECONOMICOS

1. Empresario: Deberá adquirir un servicio de hospedaje en Internet para la base de datos y para la aplicación, este hospedaje incluye el derecho al uso del software y hardware del equipo en el que se ofrezca el hospedaje.

- Hospedaje Base de datos: debe ser un servidor con sistema operativo Linux red Hat 7.3 o superior, también se requiere que tenga instalado el motor de base de datos MySQL Server versión 4.0.9 Gamma o superior. El costo anual por el hospedaje de este servidor es de \$96US.

 - Hospedaje Aplicación: debe ser un servidor con sistema operativo Windows 2000 server o superior, que tenga instalado y configurado el servidor de componentes así como también el servidor web Internet Information Server 5.0. el costo anual por el hospedaje de este servidor es de \$250US.
2. Desarrolladores: Conocimiento sobre el montaje de un sitio web: Esto encierra el conocimiento y la experiencia en el manejo de:
- Servidores web
 - Arquitectura Windows DNA
 - Desarrollo de componentes
 - Lenguaje Visual Basic 6.0
 - Amplio conocimiento sobre lenguaje HTML
 - Lenguaje XML
 - Desarrollo de páginas activas de servidor(Asp)
 - Manejo de bases de datos relacionales
 - Sistemas distribuidos

El conocimiento citado anteriormente tiene un costo de \$1'350.000.

Herramientas para el desarrollo: Para el desarrollo del proyecto se utilizarán las siguientes herramientas

COMED

Aplicación	Sistema Operativo	Licencia
Motor de base de datos MySql	Linux	GLP
Editor HTML Edit Plus 2.1	Windows	Free
Editor HTML y Asp EasyASP	Windows	Free
Editor de imágenes Lview	Windows	Free
Versión de 60 días de Visual Studio 6.0	Windows	Trial

3. Mano de obra: El costo de la mano de obra para el desarrollo de este proyecto será de \$3'250.000 este valor sale de 13 semanas, cada una a \$125.000 por persona.

4. Capacitación: La capacitación será de 15 horas, repartidas en 5 días, y el temario sería el siguiente:

1.Introducción al Aplicativo. (3 horas)

2.Conceptos generales de la arquitectura Windows DNA y sus componentes.
(3 horas)

3.Estructura del sitio web. (3 horas)

4.Conocimeinto sobre las funcionalidades de catálogo, ordenes y reportes de consumo. (3 horas)

5.Administración y ejecución del proceso de actualización y cargue de productos, inventario y clientes. (3 horas)

El costo de la capacitación será de \$225.000

5. Gastos varios: Los gastos que generan el desarrollo de la aplicación son:

Dos resmas de papel bond de 70 gramos \$ 16.000

COMED

Dos cartuchos de tinta negros: \$ 150.000
Fotocopias \$ 50.000
cd regrabable \$ 20.000
Quemada de cd regrabable \$ 20.000
Transporte \$ 100.000
Luz \$ 80.000
Teléfono \$ 100.000
Conexión Internet \$ 50.000

Para un total de \$586.000

De acuerdo a los puntos enumerados anteriormente el costo total del aplicativo será de \$ 5'411.000.

3.6 MERCADEO

El aplicativo inicialmente se comercializará en la ciudad de Bogotá orientado a las pequeñas y medianas empresas (Pymes) que comercializan productos y/o servicios que deseen ampliar su mercado y tener presencia en la web, pero con la idea futurista de cubrir todo el mercado nacional. Se hará una pre-encuesta y luego una encuesta para captar más impresiones y variables sobre el comercio que vamos a cubrir.

4. FORMULACION

¿Que tan fácil es adquirir una aplicación de comercio electrónico que sea acorde con el estilo de comercializar productos en Colombia y que además este al alcance de las pequeñas y medianas empresas de la ciudad de Bogotá?

5. DELIMITACION

La ubicación geográfica que cubriría este proyecto será la ciudad de Bogotá. El tiempo requerido para el desarrollo y la implementación de este proyecto será de 6 meses. Esta aplicación esta orientada a las pequeñas y medianas empresas de la ciudad de Bogotá que tengan por objeto social la comercialización de productos y/o servicios.

A través de Internet los clientes del empresario podrán entrar y consultar su catálogo de productos y sí no tienen podrán crearlo. También tendrán la posibilidad de montar pedidos. Así como el generar reportes tales como: listado de cliente del sitio web, cliente que más ha comprado, cliente que menos ha comprado, producto más consumido, producto menos consumido, ventas, visitas de los clientes los cuales pueden ser filtrados por rangos de fecha.

La aplicación no hará registro de información contable. Tampoco estará integrada al sistema existente en la compañía del empresario.

6. OBJETIVO GENERAL

Implementar un prototipo de comercio electrónico para pequeñas y medianas empresas de la ciudad de Bogotá que comercializan productos y/o servicios, basado en la arquitectura Windows DNA, utilizando dos sistemas operativos, uno como servidor de base de datos y otro como servidor de aplicación. El cargue masivo de información que sea necesario hacer se hará mediante el lenguaje XML.

7. OBJETIVOS ESPECIFICOS

- Realizar un estudio de mercado para determinar los clientes potenciales del aplicativo. Así como de las diferentes tecnologías que están de moda para desarrollo de aplicativos de comercio electrónico.
- Efectuar todo el montaje de hardware y software necesarios para asegurar el ambiente de desarrollo para el aplicativo.
- Diseñar y desarrollar los componentes necesarios para implementar la lógica del negocio.
- Efectuar un plan de pruebas exhaustivo sobre el aplicativo en funcionamiento, para poder realizar los ajustes necesarios.

8. JUSTIFICACION

De acuerdo a las exigencias tecnológicas del momento las pequeñas y medianas empresas de Colombia se han visto en la necesidad de actualizar sus sistemas y por ende sus estrategias comerciales, para ello deben migrar las políticas de su negocio hacia Internet.

Debido a que en el mercado del software no existen aplicativos acordes con el estilo de comercializar productos en Colombia y al alcance de las pequeñas y medianas empresas de la ciudad de Bogotá se hace necesario el desarrollo de un aplicativo que cumpla con estos requisitos. Dicho aplicativo debe estar cimentado en una arquitectura de programación sólida así como un motor de base datos robusto para que se acople a la diversidad de productos y servicios que se ofrecen en el mercado comercial colombiano dando esto solidez y garantía a la solución desarrollada.

Siendo esta una herramienta de gran utilidad tanto para el empresario como para sus clientes, para el empresario por que le permitirá descongestionar el área comercial de tareas operativas y ofrecer un valor agregado a sus clientes lo cual le da mayor solidez ante sus competidores y para los clientes por que le permitirá hacer transacciones de manera sencilla y rápida así como la posibilidad de generar reportes de consumo para poder hacer una mejora administración en sus gastos.

9. MARCO TEORICO

◆ Orientación de negocios en Internet:

Comercio Electrónico: Una posible definición de comercio electrónico sería: "cualquier forma de transacción comercial en la que las partes interactúan electrónicamente en lugar de por intercambio o contacto físico directo". Sin embargo esta definición difícilmente capta el espíritu del comercio electrónico, que en la práctica puede verse más bien como uno de esos casos en los que las necesidades de cambio y las nuevas tecnologías se aúnan para revolucionar la forma en que se llevan a cabo los negocios.

El comercio moderno está caracterizado por un incremento de la capacidad de los suministradores, de la competitividad global y de las expectativas de los consumidores. En respuesta, el comercio mundial está cambiando tanto en su organización como en su forma de actuar. Se están sobrepasando las estructuras jerárquicas antiguas y erradicando las barreras entre divisiones de empresas, así como las existentes entre las empresas y sus suministradores y clientes. Los procesos comerciales se están rediseñando de manera que atraviesen estos límites. Existen ya muchos ejemplos de procesos que afectan a una empresa entera e incluso algunos que llevan a cabo de manera conjunta las empresas y sus consumidores o suministradores.

El comercio electrónico es un medio de hacer posible y soportar tales cambios a escala global. Permite a las empresas ser más eficientes y más flexibles en sus operaciones internas, trabajar más estrechamente con sus suministradores y dar mejor respuesta a las necesidades y expectativas de sus clientes. Les permite

seleccionar los mejores proveedores, sin tener en cuenta su localización geográfica, y vender en un mercado global.

Un tipo especial de comercio electrónico es la venta electrónica, en la que un suministrador provee bienes o servicios a un cliente a cambio de un pago. Como caso especial de venta electrónica estaría aquel en el que el cliente es un consumidor ordinario en lugar de otra empresa.

Sin embargo aunque estos casos especiales tienen una considerable importancia económica, son sólo casos particulares del caso más general de cualquier forma de operación o transacción comercial llevada a cabo a través de medios electrónicos. Otros ejemplos igualmente válidos son las transacciones internas dentro de una misma empresa o el suministro de información a una organización externa con o sin cargo.

El comercio electrónico es tecnología para el cambio. Las empresas que lo miren como un "añadido" a su forma habitual de hacer negocio obtendrán sólo beneficios limitados, siendo el mayor beneficio para aquellas que sean capaces de cambiar su organización y sus procesos comerciales para explotar completamente las oportunidades ofrecidas por el comercio electrónico.

El comercio electrónico, según los agentes implicados, puede subdividirse en cuatro categorías diferentes:

empresa-empresa(B2B - Business to Business)

empresa-consumidor(B2C - Business to Customer)

empresa-administración

consumidor-administración

Empresa-empresa(B2B - Business to Business): En este caso hablamos de comercio electrónico entre empresas: clientes y proveedores. Las tecnologías de comunicación pueden aparecer en la obtención de información sobre los productos de la empresa, en la negociación de los precios, en el aprovisionamiento, en los pagos, en el intercambio de facturas, en el servicio postventa, etc.

Empresa-consumidor(B2C - Business to Customer): Es la modalidad de comercio electrónico más conocida por el público. Se busca la venta de productos finales a un consumidor.

Empresa-administración: cubre todas las transacciones entre las empresas y las organizaciones gubernamentales. Por ejemplo, en USA las disposiciones gubernamentales se publicitan en Internet y las compañías pueden responder electrónicamente. Generalmente esta categoría está empezando, pero puede crecer rápidamente si los gobiernos la usan para sus operaciones para promover la calidad y el crecimiento del comercio electrónico. Además, las administraciones pueden ofrecer también la opción del intercambio electrónico para transacciones como determinados impuestos y el pago de tasas corporativas.

Consumidor-administración: No acaba de emerger. Sin embargo, a la vez que crecen tanto las categorías empresa-consumidor y empresa-administración, los gobiernos podrán extender las interacciones electrónicas a áreas tales como los pagos de pensiones o el autoasesoramiento en devoluciones de tasas.

ANTECEDENTES:

A continuación se enumeran algunos de los aplicativos existentes en el mercado que tienen la misma orientación al aplicativo a desarrollar en este proyecto:

Commerce Server 2002 es la plataforma de Microsoft® Windows Server System™: para crear rápidamente negocios en línea de la siguiente generación. Creado con la ágil tecnología de Microsoft .NET, Commerce Server 2002 puede extender la funcionalidad de su sitio, crear rentabilidad y mejorar la experiencia del usuario en forma global. Commerce Server 2002 proporciona poderosas características para perfilar usuarios, personalizar, administrar catálogos, procesar pedidos, globalización y avanzado análisis de negocios en línea. Esta descripción destaca las capacidades de alto nivel de Commerce Server 2002.

Negocios en línea, globales e inteligentes

Commerce Server 2000 proporciona todo lo que necesita para crear y administrar una red global de clientes y socios comerciales, conducir negocios en múltiples idiomas y divisas y facultar a los usuarios de negocios a tomar decisiones de negocios más inteligentes mientras maximizan la efectividad de su negocio en línea.

Experiencia en desarrollo de la siguiente generación

Commerce Server 2002 entrega flexibilidad y velocidad óptimas para crear o extender poderosas soluciones de negocios en línea habilitados con Web Services, más rápidamente y con menos líneas de código que nunca. Un complemento natural para Microsoft Visual Studio® .NET, Commerce Server 2002 es ahora más accesible para los desarrolladores que desean aprovechar las

ventajas de sus componentes de negocios preconstruidos y las herramientas que dan soporte tanto al desarrollo de COM/ASP como Microsoft ASP.NET.

Capacidades de misión crítica

Commerce Server 2002 se crea en seis años de experiencia entregando soluciones de misión crítica, escalables de gran rendimiento, que los clientes empresariales necesitan. Numerosas soluciones en línea confiables han sido entregadas con poderosas herramientas de administración fáciles de usar que minimizan el tiempo de mantenimiento y el costo total de propiedad.

Este aplicativo tiene un costo por licencia de \$6,999 US.

IBM WebSphere Commerce Suite

Obtener millones de prospectos ya no es lo deseable. Lo deseable, es transformar los navegadores y compradores ocasionales en compradores leales, generadores de ganancias. Diseñado para ayudar a obtener sitios de entrada en la Web montados y operando rápidamente, el IBM WebSphere® Commerce Suite, Start Edition para Windows NT® y Windows 2000 lo ayuda a atraer clientes, mantener relaciones y ventas con eficacia de coste.

Características en un vistazo

- Desarrolle sitios de e-commerce que permitan que los clientes compren exactamente de la forma que desean: establezcan preferencias, naveguen en los catálogos, coloquen ítems en los carritos de compras, pasen por el cajero, seleccionen métodos de pago y despacho y paguen sus compras a través de la tarjeta de crédito.

- Administre con facilidad su sitio, tiendas o centros comerciales, mediante una interface administrativa que controla la seguridad por medio de identificación de usuario y contraseña.
- Construya y personalice su sitio con herramientas fáciles de usar, basadas en la Web.
- Cree catálogos interactivos que ayuden a vender cualquier producto (inclusive bienes durables, bienes de consumo y servicios) y conduzca los clientes a través de flujos de compras eficientes.
- Acepte y procese pagos con tarjetas de crédito mediante el procesador de pagos del IBM WebSphere Commerce.
- Alcance clientes globales mediante la entrega de informaciones basada en el lenguaje preferente y la preferencia cultural.
- Sirva a sus clientes, independiente de su localización, tiempo o dispositivo, extendiendo sus aplicativos a usuarios en aparatos inalámbricos por medio de la capacidad del e-commerce.
- Apalanque una solución basada en estándares abiertos, aceptados por la industria, como las tecnologías Java™, JavaServer Pages™, Enterprise JavaBeans™ (EJB™) y XML.

Este aplicativo tiene un costo de \$150.000 US

10. MARCO CONCEPTUAL

◆ Motor de base de datos MySql:

MySQL es el servidor de bases de datos relacionales más popular, desarrollado y proporcionado por MySQL AB. MySQL AB es una empresa cuyo negocio consiste en proporcionar servicios en torno al servidor de bases de datos MySQL. Una de las razones para el rápido crecimiento de popularidad de MySQL, es que se trata de un producto Open Source, y por lo tanto, va de la mano con este movimiento.

MySQL es un sistema de administración de bases de datos

Una base de datos es una colección estructurada de datos. La información que puede almacenar una base de datos puede ser tan simple como la de una agenda, un contador, o un libro de visitas, ó tan vasta como la de una tienda en línea, un sistema de noticias, un portal, o la información generada en una red corporativa. Para agregar, acceder, y procesar los datos almacenados en una base de datos, se necesita un sistema de administración de bases de datos, tal como MySQL.

MySQL es un sistema de administración de bases de datos relacionales

Una base de datos relacional almacena los datos en tablas separadas en lugar de poner todos los datos en un solo lugar. Esto agrega velocidad y flexibilidad. Las tablas son enlazadas al definir relaciones que hacen posible combinar datos de varias tablas cuando se necesitan consultar datos. La parte SQL de "MySQL" significa "Lenguaje Estructurado de Consulta", y es el lenguaje más usado y estandarizado para acceder a bases de datos relacionales.

MySQL es Open Source

Open Source significa que la persona que quiera puede usar y modificar MySQL. Cualquiera puede descargar el software de MySQL de Internet y usarlo sin pagar por ello. Inclusive, cualquiera que lo necesite puede estudiar el código fuente y cambiarlo de acuerdo a sus necesidades. MySQL usa la licencia GPL (Licencia Pública General GNU), para definir qué es lo que se puede y no se puede hacer con el software para diferentes situaciones. Sin embargo, si uno está incómodo con la licencia GPL o tiene la necesidad de incorporar código de MySQL en una aplicación comercial es posible comprar una versión de MySQL con una licencia comercial. Para mayor información, ver la página oficial de MySQL en la cuál se proporciona mayor información acerca de los tipos de licencias.

¿Por qué usar MySQL?

El servidor de bases de datos MySQL es muy rápido, seguro, y fácil de usar. Si eso es lo que se está buscando, se le debe dar una oportunidad a MySQL. Se pueden encontrar comparaciones de desempeño con algunos otros manejadores de bases de datos en la página de MySQL.

El servidor MySQL fue desarrollado originalmente para manejar grandes bases de datos mucho más rápido que las soluciones existentes y ha estado siendo usado exitosamente en ambientes de producción sumamente exigentes por varios años. Aunque se encuentra en desarrollo constante, el servidor MySQL ofrece hoy un conjunto rico y útil de funciones. Su conectividad, velocidad, y seguridad hacen de MySQL un servidor bastante apropiado para acceder a bases de datos en Internet.

Algunos detalles técnicos de MySQL

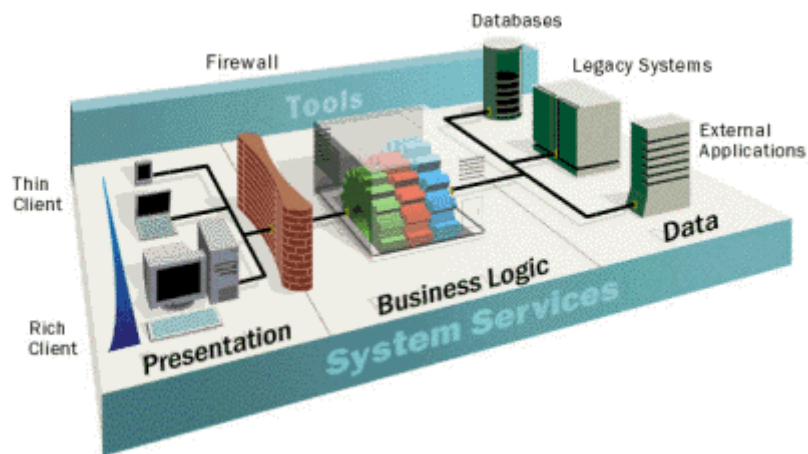
El software de bases de datos MySQL consiste de un sistema cliente/servidor que se compone de un servidor SQL multihilo, varios programas clientes y bibliotecas, herramientas administrativas, y una gran variedad de interfaces de programación (APIs). Se puede obtener también como una biblioteca multihilo que se puede enlazar dentro de otras aplicaciones para obtener un producto más pequeño, más rápido, y más fácil de manejar. Para obtener información técnica más detallada, es necesario consultar la guía de referencia de MySQL.

◆ **Arquitectura de programación Windows D.N.A.:** La plataforma Microsoft de



aplicaciones consiste en un modelo de aplicación multi-capa llamado Windows DNA y un completo conjunto de servicios de infraestructura y aplicación. Windows DNA unifica lo mejor de los servicios disponibles hoy en día en los ordenadores personales, servidores de aplicaciones y grandes

sistemas; los beneficios inherentes a los sistemas cliente – servidor, y lo mejor de las tecnologías de Internet alrededor de una arquitectura común, basada en componentes.



¿
Cuáles

son los

beneficios clave de Windows DNA?

- Windows DNA proporciona **la plataforma de servidor de aplicaciones más completa e integrada** para construir aplicaciones distribuidas. Esta plataforma incluye servicios comunes como proceso asíncrono de colas de mensajes, transacciones, servicios de componentes, activación de componentes “just-in-time”, pooling de conexiones a bases de datos, acceso a datos y publicación Web.
- **Preparado para Internet** Permite el desarrollo de soluciones que explotan completamente la flexibilidad de la plataforma de aplicaciones y el ámbito global de Internet, con capacidades de comunicación bajo demanda.
- **Respuesta al mercado más Rápida.** El desarrollo y la distribución de las aplicaciones son más rápidos, no requieren nueva formación para el desarrollador ni necesitan nuevas metodologías de construcción de software. Windows DNA expone los servicios y la funcionalidad a través de la infraestructura subyacente, de forma que se reduce la cantidad de código que deben escribir los desarrolladores.
- **Interoperatividad Real.** Incorpora interoperatividad entre todas las capas para añadir funcionalidad a los sistemas existentes. Sigue los protocolos abiertos y los estándares para poder integrar soluciones de otros fabricantes.
- **Complejidad Reducida.** Los servicios principales se encuentran integrados en el sistema operativo, y la exposición de los mismos se hace de forma uniforme mediante componentes. Esto reduce la necesidad de

que los profesionales de TI funcionen como integradores de sistemas, de manera que pueden concentrarse en resolver los problemas del negocio.

- **Independencia del Lenguaje, Herramientas y Hardware.** Suministra un modelo de componentes independientes del lenguaje, de forma que los programadores puedan usar herramientas apropiadas para cada tarea. Se basa en el modelo PC, de forma que los clientes pueden desarrollar soluciones utilizando el amplio hardware disponible.
- **Reduce el Coste Total de Propiedad.** Se consigue el desarrollo de aplicaciones sencillas de distribuir, administrar y adaptar en el tiempo.

Debido a la propia infraestructura subyacente, el modelo de presentación multi-capas, la lógica del negocio y los datos, se pueden distribuir físicamente los distintos procesos entre varios ordenadores. Sin embargo, las abstracciones principales que han funcionado en los antiguos modelos de una y dos capas – lenguajes de programación de alto nivel, sistemas de gestión de bases de datos e interfaces de usuario – no satisfacen por completo los requerimientos del desarrollo multi-capas. Se necesita un nivel distinto de abstracción para desarrollar aplicaciones escalables, gestionables y multi usuario, y en Microsoft pensamos que esta abstracción está en los componentes cooperativos.

◆ **Páginas activas de servidor (Asp):**

Páginas Active Server (ASP, Active Server Pages) es un entorno para crear y ejecutar aplicaciones dinámicas e interactivas en la Web.

Se puede combinar páginas HTML, secuencias de comandos y componentes ActiveX para crear páginas y aplicaciones Web interactivas.

El modelo de Páginas ASP

Las páginas ASP comienzan a ejecutarse cuando un usuario solicita un archivo .asp al servidor Web a través del explorador. El servidor web llama a ASP, que lee el archivo solicitado, ejecuta las secuencias de comandos que encuentre y envía los resultados al explorador del cliente. Puesto que las secuencias de comandos se ejecutan en el servidor, y NO en el cliente, es el servidor el que hace todo el trabajo necesario para generar las paginas que se envían al explorador. Las secuencias de comandos quedan ocultas a los usuarios, estos solo reciben el resultado de la ejecución en formato HTML.

Desaparece por tanto el problema de si el cliente puede o no ejecutar sentencias de comandos, el servidor Web solo envía el resultado en código HTML standard interpretable por cualquier explorador.

Crear Páginas ASP

Los archivos .asp son archivos de texto normales, no es necesario ningún editor especial para crearlos, puede usarse cualquier editor que genere código ascii.

Un archivo .asp puede contener texto, código HTML, código ASP o cualquier combinación de estos. Si no contiene código ASP se comporta como un archivo .html normal.

Nota: todos los archivos .asp requieren una parte de proceso por el servidor, por lo cual no es conveniente convertir a .asp los archivos que no contengan código.

◆ Lenguaje XML:

XML significa eXtensible markup language, o lenguaje de anotación extensible. Ya conocemos el lenguaje HTML (*hypertext markup language*), lenguaje de anotación para página webs que permite navegación tipo hipertexto; sin embargo, XML no es sólo un lenguaje, es una forma de especificar lenguajes, de ahí lo de extensible. Todo lenguaje que se exprese de una forma determinada puede ser XML. Por lo tanto, XML *no es un lenguaje para hacer mejores páginas web*, sino un lenguaje para información auto-descrita, o al menos, auto-descrita si las etiquetas están bien puestas.

XML se inició como un subconjunto de SGML (*structured generalized markup language*), un standard ISO para documentos estructurados que es sumamente complejo para poder servir documentos en la web. XML es algo así como SGML simplificado, de forma que una aplicación no necesita comprender SGML completo para interpretar un documento, sino sólo el subconjunto que se defina. Los editores SGML, sin embargo, pueden comprender XML.

Por tanto, no debe uno de pensarse que XML es para crear páginas web, o algo parecido a las página web. XML es un lenguaje que cambia el paradigma de publicación, de basado en el programa a *programación basada en el documento*. XML se puede usar para cambiar totalmente el paradigma de publicación; de un programa que recibe unas entradas y produce unas salidas, se pasa a un documento que genera otro documento, o bien programas que toman documentos y producen otros documentos. Por eso, también, y, en general, salvo en entornos de servicios web, lo normal es que el XML se use en el servidor, y se sirva otro tipo de documentos, HTML, por ejemplo, que se obtienen a base de una serie de transformaciones. Precisamente, esto hace que los documentos XML se usen dentro de entornos de aplicaciones. Este entorno de aplicaciones permite publicar

documentos XML, que, antes de ser enviados al cliente, sufrirán una serie de transformaciones para adaptarlo a los requisitos del mismo. Algunos ejemplos de entorno de aplicaciones son el Cocoon, un entorno basado en Java, libre, que permite no sólo publicar páginas XML, sino también incluir programas dentro de las páginas (XSP). No se caracteriza por su velocidad ni amigabilidad, pero es excelente como entorno de desarrollo (y el precio es inmejorable). Otra alternativa gratuita es el AxKit. Como alternativas de pago (y bien pagadas) están el Bea Weblogic, y el IBM transcoding publisher. Sobre todos estos y muchos más se trata en esta discusión en barrapunto, en la cual se menciona, por ejemplo Krysalis, un entorno de publicación basado en PHP, que incluye facilidades para ser usado a través de SOAP.

Dentro de estos entornos de desarrollo, o usándolo de cualquier otra forma, XML tiene gran número de aplicaciones. La mayor parte de los portales y sitios de noticias ya están basados en XML, porque permite estructurar la información y luego aplicarle transformaciones fáciles para presentarlo. Lo más normal es que la información esté almacenada en una base de datos, se convierta a XML y luego se transforme para servirlo al cliente. Otro ejemplo de aplicación basada en XML es la base de datos discográfica de Siniestro Total está también basada en XML, y además el código es libre. Muchos weblogs, tales como barrapunto y Slashdot, sirven sus titulares en XML (y RDF), lo cual permite procesarlo fácilmente para, por ejemplo, incluirlos en la página personal de uno (ver la barra de la derecha). Todos los sitios que sirven, o servían, páginas WAP también usan, sin otro remedio, XML. Google ofrece un interfaz de programación para acceder a sus servicios usando SOAP, un interfaz de acceso remoto que usa XML. Y se puede usar en cualquier aplicación web donde haga falta programación estructurada.

- ◆ **Visual Basic 6.0:** es un lenguaje de programación visual, también llamado lenguaje de 4ª generación. Esto quiere decir que un gran número de tareas se realizan sin escribir código, simplemente con operaciones gráficas realizadas con el ratón sobre la pantalla.

Visual Basic 6.0 es también un programa basado en objetos, aunque no orientado a objetos como C++ o Java. La diferencia está en que Visual Basic 6.0 utiliza objetos con propiedades y métodos, pero carece de los mecanismos de herencia y polimorfismo propios de los verdaderos lenguajes orientados a objetos como Java y C++.

- ◆ **Componentes DLL:**

Dll - Dynamic Link Library ("Biblioteca de vínculos dinámicos" es un archivo que contiene funciones que se pueden llamar desde aplicaciones u otras Dll. Los desarrolladores utilizan las Dll para poder reciclar el código y aislar las diferentes tareas. Las Dll no pueden ejecutarse directamente, es necesario llamarlas desde un código externo.

Definición y Explicación de un archivo .DLL

La información de este artículo esta basada en Q87934 y se aplica a:

- Microsoft Windows 98 SE (Second Edition)
- Microsoft Windows 98
- Microsoft Windows 95
- Microsoft Windows 3.0, 3.1, 3.x
- Microsoft Windows Millennium Edition (Windows ME)

SUMARIO

Una Biblioteca de Vínculos Dinámicos (DLL) file es un archivo ejecutable que permite compartir código y otros recursos para realizar ciertas tareas. Las Dll de Windows permiten que las aplicaciones puedan operar en el entorno de Windows.

MÁS INFORMACIÓN

Normalmente las Bibliotecas de Vínculos Dinámicos aparecen con la extensión ".dll"; sin embargo, ellas también pueden tener la extensión ".exe" u otra extensión. Por ejemplo, Shell.dll contiene las rutinas de arrastrar y soltar de OLE - ("Object Linking and Embedding" - *Incrustación y Vinculación de Objetos*) que pueden ser utilizadas por Windows y otros programas.

Kernel.exe, User.exe y Gdi.exe son un ejemplo de las DLLs con la extensión .exe. Estos archivos contienen código, datos o rutinas que se ejecutan en Windows. Por ejemplo, uno de estos archivos contiene la función "CreateWindow" que se utiliza cuando un programa quiere crear una nueva ventana en la pantalla.

En Windows un controlador que instalamos es también una DLL. Un programa puede abrir, habilitar, consultar, deshabilitar y cerrar un controlador basado en las instrucciones escritas en un archivo .DLL.

Las DLL se pueden encontrar en el directorio de Windows, en el directorio Windows\System o en el directorio "Archivos de programa".

Si al iniciar un programa una de sus Dll falta o está dañada podemos recibir un mensaje de error como éste: "No se encuentra xyz.dll". Si la Dll de este programa está caduca o no corresponde al estándar podemos recibir el error "Llamada al

vínculo dinámico no definido" ("Call to undefined dynalink"). En estos casos debemos obtener la Dll apropiada y colocarla en el directorio que corresponda.

Como determinar el Origen de una DLL:

Para determinar el número de versión, nombre del fabricante u otra información sobre un archivo:

1. Pulse Inicio, Buscar y pulse Archivos o carpetas.
2. Escriba el nombre del archivo que busca, por ejemplo shell32.dll, y seleccione el disco local donde quiere buscarlo y pulse "Buscar ahora".

Pulse con el botón secundario en los archivos encontrados, pulse Propiedades y pulse la pestaña Versión.

◆ Internet Information Server 5.0:

Internet Information Server es un servidor web, que incluye los servicios de: HTTP, HTTPS, FTP, SMTP (correo saliente) y NNTP (grupos de noticias). Además es capaz de ejecutar varios motores de script como: ASP, PHP, Cold Fusion, etc.

A diferencia de su hermano menor IIS 4.0, la nueva versión del Internet Information Server (IIS) 5.0 está incluida en el sistema operativo (S.O) Windows 2000 Server y Advanced Server, si no indicamos lo contrario durante la instalación del S.O esta aplicación se instalará por defecto con los servicios web y smtp, no

instalándose el servidor nntp para los grupos de noticias ni el ftp que debería especificarse si se quieren instalar.

Si por el contrario no se ha instalado IIS 5.0 es necesario ir a Mi PC -> Panel de Control -> Añadir o quitar programas y seleccionar "añadir y quitar componentes de Windows", seleccionar Internet Information Server, nos pedirá el cd de Windows 2000 para copiar e instalar el software.

Una vez instalado para acceder al panel de control del IIS 5.0 tenemos que ir a Inicio -> Programas -> Herramientas de administración -> Internet Information Server, obtendremos la MMC (Microsoft Management Console), que es el panel de control para el IIS 5.0 desde donde administraremos, instalaremos y modificaremos nuestros Webs. Por defecto cuando se instala el IIS 5.0 se crea:

- Site principal (Default Web Site)

- Site de administración, donde están las ayudas y las páginas de administración del IIS 5.0 en html, por seguridad este Site siempre se instala en otro puerto distinto al 80.

- Servidor SMTP

- Servidor NNTP para los grupos de noticias debe seleccionarse si se quiere instalar, al igual que el servidor FTP.

El IIS 5.0 por defecto se instala en los siguientes directorios:

%Systemroot%\System32\InetSrv
%Systemroot%\System32\InetSrv\IISAdmin
c:\InetPub
C:\Winnt\Help\IISHelp

El directorio del web por defecto es c:\inetpub\wwwroot

También instala dos usuarios por defecto para el acceso anónimo a los Webs:

IUSR_Nombre_de_Maquina

IWAM_Nombre_de_Maquina

Sistema Operativo Linux Red Hat :

Linux es un sistema operativo gratuito y de libre distribución inspirado en el sistema Unix, escrito por Linus Torvalds con la ayuda de miles de programadores en Internet. Unix es un sistema operativo desarrollado en 1970, una de cuyas mayores ventajas es que es fácilmente portable a diferentes tipos de ordenadores, por lo que existen versiones de Unix para casi todos los tipos de ordenadores, desde PC y Mac hasta estaciones de trabajo y superordenadores. Al contrario que otros sistemas operativos, como por ejemplo MacOS (Sistema operativo de los Apple Macintosh), Unix no está pensado para ser fácil de emplear, sino para ser sumamente flexible. Por lo tanto Linux no es en general tan sencillo de emplear como otros sistemas operativos, aunque, se están realizando grandes esfuerzos para facilitar su uso. Pese a todo la enorme flexibilidad de Linux y su gran estabilidad (y el bajo coste) han hecho de este sistema operativo una opción muy a tener en cuenta por aquellos usuarios que se dediquen a trabajar a través de redes, naveguen por Internet, o se dediquen a la programación. Además el futuro de Linux es brillante y cada vez más y más gente

y más y más empresas (entre otras IBM, Intel, Corel) están apoyando este proyecto, con lo que el sistema será cada vez más sencillo de emplear y los programas serán cada vez mejores.

◆ **Sistema Operativo Windows 2000 server:**

Microsoft Windows 2000 constituye un gran avance en las áreas de administración de redes, soporte de hardware, acceso a Internet y demás. Sin embargo, eso no quiere decir que no existan limitaciones. Windows 2000 tiene que funcionar tanto en el presente como en el futuro, y la necesidad de una compatibilidad con lo anterior conlleva unos compromisos que, aún siendo esenciales, no son probablemente lo que se podría esperar.

Sin embargo, los cambios positivos son varios. Windows 2000 es, con certeza, más fiable, más sencillo de administrar (una vez que nos hemos habituado a los cambios) y más rápido que Microsoft Windows NT. Microsoft Windows 2000 Server, llamado también Microsoft Windows 2000 Server Standard Edition, es el sucesor de Microsoft Windows NT 4. Basado en el Active Directory, incluye seguridad Kerberos e infraestructura de clave pública (PKI, Public-Key Infrastructure), Servicios de terminal (Terminal Services), COM+, Servicios de componente (Component Services), Servicios de información de Internet (Internet Information Services), Servicio de índices (Indexing Services) y Colas de mensajes (Message Queuing). También soporta hasta 4 Gb de memoria física y hasta 4 vías de multiprocesamiento simétrico (SMP, Symmetric Multiprocessing). Microsoft Windows 2000 Advanced Server es el sucesor de Microsoft Windows NT 4 Enterprise Edition. Tiene todas las características de Windows 2000 Server además de equilibrado de la carga de la red, agrupamiento de aplicaciones con gestión de fallos mejorado, hasta 8 Gb de memoria física y SMP de 8 vías.

Microsoft Windows Datacenter Server tendrá las características del Advanced Server además de SMP de 32 vías, hasta 64 Gb de memoria física y agrupamiento aún más avanzado.

Mejor protección: Todos los archivos residen en el servidor, de forma que se pueden hacer copias de seguridad como parte de los procedimientos de copia y restauración normales y centralizados.

◆ **Sistemas distribuidos:**

Los sistemas distribuidos están basados en las ideas básicas de transparencia, eficiencia, flexibilidad, escalabilidad y fiabilidad. Sin embargo estos aspectos son en parte contrarios, y por lo tanto los sistemas distribuidos han de cumplir en su diseño el compromiso de que todos los puntos anteriores sean solucionados de manera aceptable.

Transparencia: El concepto de transparencia de un sistema distribuido va ligado a la idea de que todo el sistema funcione de forma similar en todos los puntos de la red, independientemente de la posición del usuario. Queda como labor del sistema operativo el establecer los mecanismos que oculten la naturaleza distribuida del sistema y que permitan trabajar a los usuarios como si de un único equipo se tratara.

En un sistema transparente, las diferentes copias de un archivo deben aparecer al usuario como un único archivo. Queda como labor del sistema operativo el controlar las copias, actualizarlas en caso de modificación y en general, la unicidad de los recursos y el control de la concurrencia.

El que el sistema disponga de varios procesadores debe lograr un mayor rendimiento del sistema, pero el sistema operativo debe controlar que tanto los usuarios como los programadores vean el núcleo del sistema distribuido como un único procesador. El paralelismo es otro punto clave que debe controlar el sistema operativo, que debe distribuir las tareas entre los distintos procesadores como en un sistema multiprocesador, pero con la dificultad añadida de que ésta tarea hay que realizarla a través de varios ordenadores.

Eficiencia: La idea base de los sistemas distribuidos es la de obtener sistemas mucho más rápidos que los ordenadores actuales. Es en este punto cuando nos encontramos de nuevo con el paralelismo.

Para lograr un sistema eficiente hay que descartar la idea de ejecutar un programa en un único procesador de todo el sistema, y pensar en distribuir las tareas a los procesadores libres más rápidos en cada momento.

La idea de que un procesador vaya a realizar una tarea de forma rápida es bastante compleja, y depende de muchos aspectos concretos, como la propia velocidad del procesador, pero también la localidad del procesador, los datos, los dispositivos, etc. Se han de evitar situaciones como enviar un trabajo de impresión a un ordenador que no tenga conectada una impresora de forma local.

Flexibilidad: Un proyecto en desarrollo como el diseño de un sistema operativo distribuido debe estar abierto a cambios y actualizaciones que mejoren el funcionamiento del sistema. Esta necesidad ha provocado una diferenciación entre las dos diferentes arquitecturas del núcleo del sistema operativo: el núcleo

monolítico y el micronúcleo. Las diferencias entre ambos son los servicios que ofrece el núcleo del sistema operativo. Mientras el núcleo monolítico ofrece todas las funciones básicas del sistema integradas en el núcleo, el micronúcleo incorpora solamente las fundamentales, que incluyen únicamente el control de los procesos y la comunicación entre ellos y la memoria. El resto de servicios se cargan dinámicamente a partir de servidores en el nivel de usuario.

Núcleo monolítico: Como ejemplo de sistema operativo de núcleo monolítico está UNIX. Estos sistemas tienen un núcleo grande y complejo, que engloba todos los servicios del sistema. Está programado de forma no modular, y tiene un rendimiento mayor que un micronúcleo. Sin embargo, cualquier cambio a realizar en cualquier servicio requiere la parada de todo el sistema y la recompilación del núcleo.

Micronúcleo: La arquitectura de micronúcleo ofrece la alternativa al núcleo monolítico. Se basa en una programación altamente modular, y tiene un tamaño mucho menor que el núcleo monolítico. Como consecuencia, el refinamiento y el control de errores son más rápidos y sencillos. Además, la actualización de los servicios es más sencilla y ágil, ya que sólo es necesaria la recompilación del servicio y no de todo el núcleo. Como contraprestación, el rendimiento se ve afectado negativamente.

En la actualidad la mayoría de sistemas operativos distribuidos en desarrollo tienden a un diseño de micronúcleo. Los núcleos tienden a contener menos errores y a ser más fáciles de implementar y de corregir. El sistema pierde ligeramente en rendimiento, pero a cambio consigue un gran aumento de la flexibilidad.

Escalabilidad: Un sistema operativo distribuido debería funcionar tanto para una docena de ordenadores como varios millares. Igualmente, debería no ser determinante el tipo de red utilizada (LAN o WAN) ni las distancias entre los equipos, etc.

Aunque este punto sería muy deseable, puede que las soluciones válidas para unos cuantos ordenadores no sean aplicables para varios miles. Del mismo modo el tipo de red condiciona tremendamente el rendimiento del sistema, y puede que lo que funcione para un tipo de red, para otro requiera un nuevo diseño.

La escalabilidad propone que cualquier ordenador individual ha de ser capaz de trabajar independientemente como un sistema distribuido, pero también debe poder hacerlo conectado a muchas otras máquinas.

Fiabilidad: Una de las ventajas claras que nos ofrece la idea de sistema distribuido es que el funcionamiento de todo el sistema no debe estar ligado a ciertas máquinas de la red, sino que cualquier equipo pueda suplir a otro en caso de que uno se estropee o falle.

La forma más evidente de lograr la fiabilidad de todo el sistema está en la redundancia. La información no debe estar almacenada en un solo servidor de archivos, sino en por lo menos dos máquinas. Mediante la redundancia de los principales archivos o de todos evitamos el caso de que el fallo de un servidor bloquee todo el sistema, al tener una copia idéntica de los archivos en otro equipo.

Otro tipo de redundancia más compleja se refiere a los procesos. Las tareas críticas podrían enviarse a varios procesadores independientes, de forma que el primer procesador realizaría la tarea normalmente, pero ésta pasaría a ejecutarse en otro procesador si el primero hubiera fallado.

Comunicación: La comunicación entre procesos en sistemas con un único procesador se lleva a cabo mediante el uso de memoria compartida entre los procesos. En los sistemas distribuidos, al no haber conexión física entre las distintas memorias de los equipos, la comunicación se realiza mediante la transferencia de mensajes.

11. MARCO LEGAL

◆ Ley de comercio electrónico:

Ley Número 527 de 1999

(agosto 18)

Por medio de la cual se define y reglamenta el acceso y uso de los mensajes de datos, del comercio electrónico y de las firmas digitales, y se establecen las entidades de certificación y se dictan otras disposiciones.

El Congreso de Colombia

DECRETA :

PARTE I

PARTE GENERAL

CAPÍTULO I

Disposiciones Generales

Artículo 1°. Ámbito de aplicación. La presente ley será aplicable a todo tipo de información en forma de mensaje de datos, salvo en los siguientes casos:

- a) En las obligaciones contraídas por el Estado colombiano en virtud de Convenios o Tratados internacionales.
- b) En las advertencias escritas que por disposición legal deban ir necesariamente impresas en cierto tipo de productos en razón al riesgo que implica su comercialización, uso o consumo.

Artículo 2°. Definiciones. Para los efectos de la presente ley se entenderá por:

- a) **Mensaje de Datos.** La información generada, enviada, recibida, almacenada o comunicada por medios electrónicos, ópticos o similares, como pudieran ser, entre otros, el Intercambio Electrónico de Datos (EDI), Internet, el correo electrónico, el telegrama, el télex o el telefax
- b) **Comercio electrónico.** Abarca las cuestiones suscitadas por toda relación de índole comercial, sea o no contractual, estructurada a partir de la utilización de uno o más mensajes de datos o de cualquier otro medio similar. Las relaciones de índole comercial comprenden, sin limitarse a ellas, las siguientes operaciones: toda operación comercial de suministro o intercambio de bienes o servicios; todo acuerdo de distribución; toda operación de representación o mandato comercial; todo tipo de operaciones financieras, bursátiles y de seguros; de construcción de obras; de consultoría; de ingeniería; de concesión de licencias; todo acuerdo de concesión o explotación de un servicio público; de empresa conjunta y otras

formas de cooperación industrial o comercial; de transporte de mercancías o de pasajeros por vía aérea, marítima y férrea, o por carretera;

c) **Firma Digital**. Se entenderá como un valor numérico que se adhiere a un mensaje de datos y que, utilizando un procedimiento matemático conocido, vinculado a la clave del iniciador y al texto del mensaje, permite determinar que este valor se ha obtenido exclusivamente con la clave del iniciador y que el mensaje inicial no ha sido modificado después de efectuada la transformación;

d) **Entidad de Certificación**. Es aquella persona que, autorizada conforme a la presente Ley, está facultada para emitir certificados en relación con las firmas digitales de las personas, ofrecer o facilitar los servicios de registro y estampado cronológico de la transmisión y recepción de mensajes de datos, así como cumplir otras funciones relativas a las comunicaciones basadas en las firmas digitales.

e) **Intercambio Electrónico de Datos (EDI)**. La transmisión electrónica de datos de una computadora a otra, que está estructurada bajo normas técnicas convenidas al efecto;

f) **Sistema de Información**. Se entenderá todo sistema utilizado para generar, enviar, recibir, archivar o procesar de alguna otra forma mensajes de datos.

Artículo 3°. *Interpretación.* En la interpretación de la presente ley habrán de tenerse en cuenta su origen internacional, la necesidad de promover la uniformidad de su aplicación y la observancia de la buena fe.

Las cuestiones relativas a materias que se rijan por la presente ley y que no estén expresamente resueltas en ella, serán dirimidas de conformidad con los principios generales en que ella se inspira.

Artículo 4°. *Modificación mediante acuerdo.* Salvo que se disponga otra cosa, en las relaciones entre partes que generan, envían, reciben, archivan o procesan de alguna otra forma mensajes de datos, las disposiciones del Capítulo III, Parte I, podrán ser modificadas mediante acuerdo.

Artículo 5°. *Reconocimiento jurídico de los mensajes de datos.* No se negarán efectos jurídicos, validez o fuerza obligatoria a todo tipo de información por la sola razón de que esté en forma de mensaje de datos.

CAPÍTULO II

Aplicación de los requisitos jurídicos de los mensajes de datos

Artículo 6°. *Escrito.* Cuando cualquier norma requiera que la información conste por escrito, ese requisito quedará satisfecho con un mensaje de datos, si la información que éste contiene es accesible para su posterior consulta.

Lo dispuesto en este artículo se aplicará tanto si el requisito establecido en cualquier norma constituye una obligación, como si las normas prevén consecuencias en el caso de que la información no conste por escrito.

Artículo 7º. Firma. Cuando cualquier norma exija la presencia de una firma o establezca ciertas consecuencias en ausencia de la misma, en relación con un mensaje de datos, se entenderá satisfecho dicho requerimiento si:

- a) Se ha utilizado un método que permita identificar al iniciador de un mensaje de datos y para indicar que el contenido cuenta con su aprobación.
- b) Que el método sea tanto confiable como apropiado para el propósito por el cual el mensaje fue generado o comunicado.

Lo dispuesto en este artículo se aplicará tanto si el requisito establecido en cualquier norma constituye una obligación, como si las normas simplemente prevén consecuencias en el caso de que no exista una firma.

Artículo 8º. Original. Cuando cualquier norma requiera que la información sea presentada y conservada en su forma original, ese requisito quedará satisfecho con un mensaje de datos, si:

- a) Existe alguna garantía confiable de que se ha conservado la integridad de la información, a partir del momento en que se generó por primera vez en su forma definitiva, como mensaje de datos o en alguna otra forma;
- b) De requerirse que la información sea presentada, si dicha información puede ser mostrada a la persona que se deba presentar.

Lo dispuesto en este artículo se aplicará tanto si el requisito establecido en cualquier norma constituye una obligación, como si las normas simplemente prevén consecuencias en el caso de que la información no sea presentada o conservada en su forma original.

Artículo 9º. Integridad de un mensaje de datos. Para efectos del artículo anterior, se considerará que la información consignada en un mensaje de datos es íntegra, si ésta ha permanecido completa e inalterada, salvo la adición de algún endoso o de algún cambio que sea inherente al proceso de comunicación, archivo o presentación. El grado de confiabilidad requerido, será determinado a la luz de los fines para los que se generó la información y de todas las circunstancias relevantes del caso.

Artículo 10. Admisibilidad y fuerza probatoria de los mensajes de datos. Los mensajes de datos serán admisibles como medios de prueba y su fuerza probatoria es la otorgada en las disposiciones del Capítulo VIII del Título XIII, Sección Tercera, Libro Segundo del Código de Procedimiento Civil.

En toda actuación administrativa o judicial, no se negará eficacia, validez o fuerza obligatoria y probatoria a todo tipo de información en forma de un mensaje de datos, por el sólo hecho que se trate de un mensaje de datos o en razón de no haber sido presentado en su forma original.

Artículo 11. Criterio para valorar probatoriamente un mensaje de datos. Para la valoración de la fuerza probatoria de los mensajes de datos a que se refiere esta ley, se tendrán en cuenta las reglas de la sana crítica y demás criterios

reconocidos legalmente para la apreciación de las pruebas. Por consiguiente habrán de tenerse en cuenta: la confiabilidad en la forma en la que se haya generado, archivado o comunicado el mensaje, la confiabilidad en la forma en que se haya conservado la integridad de la información, la forma en la que se identifique a su iniciador y cualquier otro factor pertinente.

Artículo 12. *Conservación de los mensajes de datos y documentos.* Cuando la Ley requiera que ciertos documentos, registros o informaciones sean conservados, ese requisito quedará satisfecho, siempre que se cumplan las siguientes condiciones:

1. Que la información que contengan sea accesible para su posterior consulta;
2. Que el mensaje de datos o el documento sea conservado en el formato en que se haya generado, enviado o recibido o en algún formato que permita demostrar que reproduce con exactitud la información generada, enviada o recibida, y
3. Que se conserve, de haber alguna, toda información que permita determinar el origen, el destino del mensaje, la fecha y la hora en que fue enviado o recibido el mensaje o producido el documento.

No estará sujeta a la obligación de conservación, la información que tenga por única finalidad facilitar el envío o recepción de los mensajes de datos.

Los libros y papeles del comerciante podrán ser conservados en cualquier medio técnico que garantice su reproducción exacta.

Artículo 13. *Conservación de mensajes de datos y archivo de documentos a través de terceros.* El cumplimiento de la obligación de conservar documentos, registros o informaciones en mensajes de datos, se podrá realizar directamente o a través de terceros, siempre y cuando se cumplan las condiciones enunciadas en el artículo anterior.

CAPÍTULO III

Comunicación de los mensajes de datos

Artículo 14. *Formación y validez de los contratos.* En la formación del contrato, salvo acuerdo expreso entre las partes, la oferta y su aceptación podrán ser expresadas por medio de un mensaje de datos. No se negará validez o fuerza obligatoria a un contrato por la sola razón de haberse utilizado en su formación uno o más mensajes de datos.

Artículo 15. *Reconocimiento de los mensajes de datos por las partes.* En las relaciones entre el iniciador y el destinatario de un mensaje de datos, no se negarán efectos jurídicos, validez o fuerza obligatoria a una manifestación de voluntad u otra declaración por la sola razón de haberse hecho en forma de mensaje de datos.

Artículo 16. *Atribución de un mensaje de datos.*- Se entenderá que un mensaje de datos proviene del iniciador, cuando éste ha sido enviado por:

1. El propio iniciador.
2. Por alguna persona facultada para actuar en nombre del iniciador respecto de ese mensaje, o
3. Por un sistema de información programado por el iniciador o en su nombre para que opere automáticamente.

Artículo 17. *Presunción del origen de un mensaje de datos.* Se presume que un mensaje de datos ha sido enviado por el iniciador, cuando:

1. Haya aplicado en forma adecuada el procedimiento acordado previamente con el iniciador, para establecer que el mensaje de datos provenía efectivamente de éste, o
2. El mensaje de datos que reciba el destinatario resulte de los actos de una persona cuya relación con el iniciador, o con algún mandatario suyo, le haya dado acceso a algún método utilizado por el iniciador para identificar un mensaje de datos como propio.

Artículo 18. *Concordancia del mensaje de datos enviado con el mensaje de datos recibido.* Siempre que un mensaje de datos provenga del iniciador o que se entienda que proviene de él, o siempre que el destinatario tenga derecho a actuar con arreglo a este supuesto, en las relaciones entre el iniciador y el destinatario, éste último tendrá derecho a considerar que el mensaje de datos recibido corresponde al que quería enviar el iniciador, y podrá proceder en consecuencia. El destinatario no gozará de este derecho si sabía o hubiera sabido, de haber actuado con la debida diligencia o de haber aplicado algún método convenido, que la transmisión había dado lugar a un error en el mensaje de datos recibido.

Artículo 19. *Mensajes de datos duplicados.* Se presume que cada mensaje de datos recibido es un mensaje de datos diferente, salvo en la medida en que duplique otro mensaje de datos, y que el destinatario sepa, o debiera saber, de haber actuado con la debida diligencia o de haber aplicado algún método convenido, que el nuevo mensaje de datos era un duplicado.

Artículo 20. *Acuse de recibo.* Si al enviar o antes de enviar un mensaje de datos, el iniciador solicita o acuerda con el destinatario que se acuse recibo del mensaje de datos, pero no se ha acordado entre éstos una forma o método determinado para efectuarlo, se podrá acusar recibo mediante:

- a) Toda comunicación del destinatario, automatizada o no, o
- b) Todo acto del destinatario que baste para indicar al iniciador que se ha recibido el mensaje de datos.

Si el iniciador ha solicitado o acordado con el destinatario que se acuse recibo del mensaje de datos, y expresamente aquél ha indicado que los efectos del mensaje de datos estarán condicionados a la recepción de un acuse de recibo, se considerará que el mensaje de datos no ha sido enviado en tanto que no se haya recepcionado el acuse de recibo.

Artículo 21. *Presunción de recepción de un mensaje de datos.* Cuando el iniciador recepcione acuse recibo del destinatario, se presumirá que éste ha recibido el mensaje de datos.

Esa presunción no implicará que el mensaje de datos corresponda al mensaje recibido. Cuando en el acuse de recibo se indique que el mensaje de datos recepcionado cumple con los requisitos técnicos convenidos o enunciados en alguna norma técnica aplicable, se presumirá que ello es así.

Artículo 22. *Efectos jurídicos.* Los artículos 20 y 21 únicamente rigen los efectos relacionados con el acuse de recibo. Las consecuencias jurídicas del mensaje de datos se regirán conforme a las normas aplicables al acto o negocio jurídico contenido en dicho mensaje de datos.

Artículo 23. *Tiempo del envío de un mensaje de datos.* De no convenir otra cosa el iniciador y el destinatario, el mensaje de datos se tendrá por expedido cuando ingrese en un sistema de información que no esté bajo control del iniciador o de la persona que envió el mensaje de datos en nombre de éste.

Artículo 24. *Tiempo de la recepción de un mensaje de datos.* De no convenir otra cosa el iniciador y el destinatario, el momento de la recepción de un mensaje de datos se determinará como sigue:

a. Si el destinatario ha designado un sistema de información para la recepción de mensaje de datos, la recepción tendrá lugar:

1. En el momento en que ingrese el mensaje de datos en el sistema de información designado; o

2. De enviarse el mensaje de datos a un sistema de información del destinatario que no sea el sistema de información designado, en el momento en que el destinatario recupere el mensaje de datos;

b. Si el destinatario no ha designado un sistema de información, la recepción tendrá lugar cuando el mensaje de datos ingrese a un sistema de información del destinatario.

Lo dispuesto en este artículo será aplicable aun cuando el sistema de información esté ubicado en lugar distinto de donde se tenga por recibido el mensaje de datos conforme al artículo siguiente.

Artículo 25. *Lugar del envío y recepción del mensaje de datos.* De no convenir otra cosa el iniciador y el destinatario, el mensaje de datos se tendrá por expedido en el lugar donde el iniciador tenga su establecimiento y por recibido en el lugar donde el destinatario tenga el suyo. Para los fines del presente artículo:

a. Si el iniciador o destinatario tienen más de un establecimiento, su establecimiento será el que guarde una relación más estrecha con la operación subyacente o, de no haber una operación subyacente, su establecimiento principal.

- b. Si el iniciador o el destinatario no tienen establecimiento, se tendrá en cuenta su lugar de residencia habitual.

PARTE IV

REGLAMENTACIÓN Y VIGENCIA

Artículo 45. La Superintendencia de Industria y Comercio contará con un término adicional de doce (12) meses, contados a partir de la publicación de la presente ley, para organizar y asignar a una de sus dependencias la función de inspección, control y vigilancia de las actividades realizadas por las entidades de certificación, sin perjuicio de que el Gobierno Nacional cree una unidad especializada dentro de ella para tal efecto.

Artículo 46. *Prevalencia de las leyes de protección al consumidor.* La presente Ley se aplicará sin perjuicio de las normas vigentes en materia de protección al consumidor.

Artículo 47. Vigencia y Derogatorias. La presente ley rige desde la fecha de su publicación y deroga las disposiciones que le sean contrarias.

12. HIPOTESIS

12.1 GENERAL:

¿La arquitectura y estructura planteadas serán las más apropiadas para obtener mejores tiempos de respuesta en Internet?

12.2 DE TRABAJO:

¿La estructura del modelo entidad relación será lo suficientemente flexible para ser adoptada por los diferentes negocios de las pequeñas y medianas empresas de Bogotá?

¿El lenguaje XML soportará transacciones con grandes volúmenes de información?

¿La información contenida en los reportes será la esperada por los clientes?

¿Que tan intuitivo será el manejo de las funcionalidades que conforman el aplicativo?

13. METODOLOGÍA

13.1 TIPO DE INVESTIGACIÓN: se ubica en el tipo de investigación cuasiexperimental.

13.2 LÍNEAS DE INVESTIGACIÓN: Este proyecto esta enmarcado dentro de las siguientes líneas de investigación: Redes y sistemas distribuidos e ingeniería de software.

1. Para la ejecución de esta etapa se hizo uso del conocimiento del proceso de ordenes de la empresa Ofixpres S.A.

Para el montaje de una orden en Ofixpres se siguen los siguientes pasos:

1. Recibir el fax o el archivo por correo electrónico de la orden a montar.
2. El representante de servicio al cliente entra al sistema y con base en la orden recibida del cliente comienza a digitar esta.
3. Si se presenta algún faltante el representante de servicio cliente llama al cliente para contarle la situación y preguntarle si se cambia el producto o se deja el faltante para su posterior envío.
4. Luego de esto se genera un documento sobre el pedido conocido como envío, el cual permite que la información del pedido pase a bodega.
5. Allí se genera unos listados conocidos como las olas, esto es un conjunto de ordenes de diferentes cliente los cuales deben ser

separados y empacados, a este proceso se le conoce con el nombre de picking.

6. El siguiente paso es el cargue de las ordenes a los camiones repartidores, obviamente existen pedidos nacionales y locales.
7. Lo último es la entrega al cliente.

Como se puede apreciar en el proceso de una orden en la empresa Ofixpres las áreas que intervienen son las siguientes:

- Servicio al cliente
- Bodega y logística

Como se puede ver en el proceso expuesto anteriormente un cliente tendría que esperar a que el representante de servicio al cliente verifique las existencias de cada uno de los productos que conforman la orden para poder identificar cuales de estos generan faltantes, con el desarrollo de este proyecto el cliente de manera inmediata podrá identificar dichos faltantes y así tomar la decisión mas acertada para su necesidad.

13.3. ETAPAS O FASES:

13.3.1 Etapa Preliminar: Para la ejecución de esta etapa se hizo necesario realizar un estudio de mercado en la ciudad de Bogotá orientado a las pequeñas y medianas empresas (Pymes) que comercializan productos y/o servicios que deseen ampliar su mercado y tener presencia en la web, pero con la idea futurista

de cubrir todo el mercado nacional. Se recopilaron antecedentes, pre-encuesta y luego una encuesta para captar más impresiones y variables sobre el comercio que vamos a cubrir.

Recopilación de Antecedentes:

- Información estadística: La información recopilada son las estadísticas del DANE, INE – Instituto Nacional de Estadísticas, Cámara de Comercio (Anexo Cuadros Estadísticos 1)
- Características del mercado: Todas las medianas y pequeñas empresas de la ciudad de Bogotá, que tengan por objeto comercializar productos y/o servicios.
- Usos y especificaciones del bien o servicio: El presente proyecto tiene como objeto el desarrollo de un aplicativo de orientación web, el cual permitirá que un empresario comercialice su producto y/o servicios.
 - **Lo usan:** todas las pequeñas y medianas empresas que comercializan y desean tener presencia en Internet.
 - **Para que lo usan:** - Para agilizar el proceso de venta
 - Para ampliar el mercado
 - Dar a conocer la compañía
- Tipo de Idiosincrasia: Va dirigido al consumidor final, ya que es un producto comercial y esta alcance de una empresa pública o privada.
- Publicidad:
 - Proveer una demostración del aplicativo en forma de un demo que será entregado a diferentes microempresas en la ciudad de Bogotá.

- Publicitar por Internet este producto dando a conocer sus ventajas y alcances
- Distribuidores reconocidos de hardware se les obsequia en versiones instaladas el aplicativo y sus productos.
- Realizar una pequeña pre-encuesta con el fin de obtener una primera información acerca del mercado a estudiar, la cual se va a dirigir a las empresas PYMES al azar. (Anexo pre-encuesta N.2)
- Promover nuestro aplicativo en las revistas y periódicos como es la revista semana, el tiempo.

Delimitaciones Geográficas:

El aplicativo inicialmente se comercializara en Bogotá, pero con la idea futurista de cubrir todo el mercado nacional, dirigido a los estratos 3, 4 y 5 a usuarios de pequeñas y medianas empresas que estén interesados en ampliar su mercado de bienes o servicios.

Muestreo estadístico:

La muestra será de 50 empresas medianas y pequeñas empresas de los diferentes sectores de la ciudad de Bogotá.

FICHA TÉCNICA

☀ Cobertura a 50 PYMES en la ciudad de Bogotá, estratos 3, 4 y 5 ciudad de Bogotá.

☀ Muestreo al azar:

Se realizó la encuesta a la personas de 20 a 50 años

Nivel de seguridad 90%

Zonas de Encuesta: Norte, Sur, Oriente y Occidente

☀ Resultados de la encuesta: La muestra estadística arrojó unos resultados positivos mostrando gran interés por el aplicativo.

Por esto para la primera pregunta **nos ubica la zona** de la empresa al igual que su dirección así; (Ver anexo No. 3)

un 24% = se aplico la encuesta a la zona sur y norte

un 20% a la zona oriente y occidente

y un 12% corresponde a las zonas nororiente, noroccidente, suroriente, suroccidente etc.

Para la segunda pregunta: **Nivel de Ingreso Mensual por ventas o servicios**: el 25 de las personas encuestadas tenían conocimiento de las ventas y a pesar de que las ventas han disminuido, quieren aumentar sus ventas, aunque el estudio nos demuestra que no les ha ido mal ya que se ubican entre 2.100.000 a más (ver anexo No. 4)

Para la tercera pregunta: **Comercializa Productos o servicios**. Los resultados arrojados son que la mayoría de las empresas venden productos (38) y en una menor escala son empresas dedicadas a prestar servicios (Ver anexo No. 5)

Para la cuarta pregunta: **Le interesaría tener presencia en Internet** la mayoría de la gente esta interesada en tener presencia en internet, porque es un mundo

virtual con gran acogida en el mercado, es una nueva tecnología y no se puede quedar atrás y por ultimo (Ver Anexo No. 6)

Quinta pregunta: **Le gustaría mejorar su forma de comercializar en Base de Datos, Clientes, Aplicativos, Productos.** En esta pregunta los encuestados tenían la opción de marcar dos o más opciones , quedando así: a los empresarios les gustaría mejorar su forma de comercializar en los que se refiere a productos y clientes(Ver anexo No. 7

El análisis demostró que una gran mayoría necesita mejorar su forma de comercializar, ya sea para el aplicativo, base de datos, productos, clientes, y así se le prestará al cliente un servicio de intermediación en donde por medio de esta empresa realizara la compra del producto elegido.

Análisis de Demanda:

- Nivel de Ingreso: El promedio general de las pequeñas y medianas empresas del país acostumbran a dejar presupuesto anual entre 1.000.000.00 a 2.000.000.00 para invertir en tecnología.

A) Ingreso por ventas

Año 1	Año 2	Año 3
\$10.000.000	\$12.000.000	\$7.000.000.00

B) Clientes

Año 1	Año 2	Año 3
-------	-------	-------

1.824	2.000	4.000
-------	-------	-------

- Cambio en la distribución geográfica: Si la competencia es muy alta en la ciudad de Bogotá (ubicación inicial) se podrá ampliar el mercado incursionando en las ciudades de Cali, Medellín, Barranquilla y posteriormente en las demás regiones del país.
- Cambio en los gustos: Debido a la naturaleza de este proyecto estos cambios no afectan la demanda.
- Innovación Tecnológica: Se puede ver afectado por los aplicativos de última generación del ramo del Comercio Electrónico con la gran ventaja de que entre más nueva tecnología es más costoso.
- Análisis de la demanda actual: Para poder tener un conocimiento del mercado y sus indicadores tomamos como base las investigaciones probabilistas descriptiva, ya que proporciona indicadores de desarrollo de la actividad comercial y así podemos obtener un conocimiento del comportamiento de las actividades económicas del comercio en generales que permita el análisis de su estructura y la conformación de los agregados macroeconómicos.
- Proyección de la demanda: Para esta fase estamos en una etapa preliminar es decir recogiendo información y analizando cada una de esas variables, ya que se debe cuantificar y caracterizar las condiciones de vida de los empresarios, así también el aumento de las nuevas tecnologías y las variables que conllevan.

Conclusiones del Estudio de Mercado:

El análisis demostró que una gran mayoría necesita mejorar su forma de comercializar, ya sea para el aplicativo, base de datos, productos, clientes, y así se le prestará al cliente un servicio de intermediación en donde por medio de esta empresa realizara la compra del producto elegido.

13.3.2 Efectuar montaje de hardware y software: Este montaje esta desarrollado con la plataforma Windows DNA, y se divide en dos capas una de datos y la otra de aplicación y presentación:

- **Capa de datos:** Esta capa es la que hacer referencia al servidor de base de datos, el cual debe tener sistema operativo Linux Red Hat 7.3 o superior y MySql server versión 4.0.9 gamma.

- **Capa de aplicación y presentación:** En esta capa encontramos un servidor con sistema operativo Windows 2000 server o superior, que tenga instalado y configurado el servidor web Internet Information Server 5.0 o superior, así como el servidor de componente COM, además de tener los componentes de la aplicación, este servidor también contendrá las imágenes, presentaciones en flash, páginas Asp y html que hacen la interfaz con el usuario final.

Este montaje se puede apreciar con mayor claridad en el diagrama siguiente: (Ver Anexo No. 8)

13.3.3 Plan de pruebas: Las pruebas que se realizaron en el aplicativo son:

- Prueba de validación: En cada uno de los campos de las páginas se digitaron datos para probar las respectivas validaciones.
- Prueba del Sistema. Se realizaron pruebas del aplicativo en los Sistemas Operativos de WIN98 y WIN2000 para ver las fallas que pudiera presentar y una de las que detectamos fue la exportación en excel.
- Muestreo: De un grupo de personas se seleccionaron algunos estudiantes, técnicos, tecnólogos y profesionales para que analizaran el software y dieran a conocer tanto lo negativo como lo positivo, de acuerdo a ello, se revisa las observaciones y las acertadas o coherentes para hacer las respectivas correcciones.

13.3.4. Diseño:

Para efectuar el diseño del aplicativo se estipularon unos estándares para el nombre de los diferentes elementos (base de datos, tablas, campos, variables, clases, métodos, funciones, objetos, etc.).

Estándares:

➤ **Estándares para estructura al nivel de filesystem**

Para el manejo del aplicativo de Comed se maneja la siguiente estructura:

Carpeta General:	Directorio raíz del sitio
Subcarpeta de sitio:	Directorio raíz/ayuda
Subcarpeta de sitio:	Directorio raíz/imágenes
Subcarpeta de sitio:	Directorio raíz/scripts
Subcarpeta de sitio:	Directorio raíz/xml
Subcarpeta de sitio:	Directorio raíz/reportes
Subcarpeta de sitio:	Directorio raíz/logs
Subcarpeta de sitio:	Directorio raíz/backupdb

➤ Estándares para el manejo de documentos

Títulos:

Fuente:	Arial
Tamaño:	16 puntos
Color:	Negro
Estilo:	Negrilla

Subtítulos:

Fuente:	Arial
Tamaño:	14 puntos
Color:	Negro
Estilo:	Negrilla

El cuerpo del texto:

Fuente:	Arial
Tamaño:	12 puntos
Color:	Negro
Estilo:	Normal

➤ Estandarización para variables

Objetos en aplicativo Sistema de Conversión de datos a XML:

Se tomarán las tres primeras letras consonantes para identificar cada objeto utilizado en el aplicativo seguidas de un sustantivo alusivo al uso que tendrá el mismo.

Ejemplo:

El nombre formulario del inicio de la aplicación quedaría así:

frmcontacto

El botón de guardar el archivo XML quedaría así:

btnxml

➤ **Base de datos:**

- Nombre de la base de datos: se colocan las letras db y luego el nombre de la sigla del proyecto
- Nombre tablas: se coloca el mismo nombre de la entidad del MER, en singular y además con una longitud máxima de 12 caracteres, en el caso que sea una tabla que recibe relación de dos tablas se le coloca como nombre las cuatro primeras letras del nombre de una tabla luego la letra x y luego de esto las cuatro primeras letras del nombre de la otra tabla.

Ejemplo:

Entidad :Ciudad
Nombre Tabla: ciudad

Entidad: relaciona catalogo con producto
Nombre Tabla: cataxprod

- Nombre campos: para este caso se colocan las cuatro primeras letras de la tabla seguido de las cuatro primeras letras del nombre del campo, en el caso de que el nombre del campo sea compuesto por dos palabras entonces se colocan las cuatro primeras letras del nombre de la tabla seguido de las dos primeras letras de la primera

palabra del nombre del campo seguido de las dos primeras letras del segundo nombre del campo.

Ejemplo:

Campo: Código

Tabla: Ciudad

El nombre de la tabla quedaría así: **ciudcodi**

Campo: Fecha Creación

Tabla: Cliente

El nombre del campo quedaría así: **cliefecr**

➤ **Clases:**

Los nombres de las clases son definidos de acuerdo al objetivo principal que esta tenga, pero a su nombre se le antepone las letras cls y la primera letra del nombre de la clase va en letra mayúscula.

Ejemplo:

Clase de Asp para pintar datos de una tabla, el nombre quedaría así:

clsAsp

➤ **Métodos:**

Los nombre de los métodos sin importar a que pertenezcan es decir a clases o a módulos, se asigna con el verbo de la acción a realizar en infinito acompañado de una sustantivo el cual es el que recibe la acción del verbo.

Ejemplo:

Existe un método en la clase clsAsp que permite pintar el contenido de una matriz y su nombre es **PintarMatriz**.

➤ **Funciones**

Los nombres de las funciones se asigna de manera similar a la de los métodos, es decir se usa un verbo en infinitivo correspondiente a la acción a realizar y va acompañado de un sustantivo que es quien recibe la acción del verbo.

Ejemplo:

Existe una función en la clase clsCrypto que permite encriptar cadenas y su nombre es **EncriptarCadena**

➤ **Procedimientos:**

La asignación del nombre de un procedimiento se hace mediante un verbo en infinitivo correspondiente a la acción a ejecutar y va acompañado de un sustantivo que recibe dicha acción.

Ejemplo:

Procedimiento para crear una conexión a una base de datos, el nombre quedaría así:

CrearConexión

Objetos en las páginas Asp:

Para asignar nombre a estos se utiliza las letras obj y luego el nombre del objeto que hace referencia a la clase de la cual se instancia.

Ejemplo:

Clase: Comed.clsCrypto

Objeto: objCrypto

➤ **Convenciones nombres variables:**

Para asignar el nombre a las variables se tiene en cuenta el tipo de dato y obviamente lo que va a contener, entonces se coloca la abreviatura según el tipo de dato y luego el nombre de lo que va a contener dejando en mayúscula la primera letra de esta palabra. Para el caso del tipo de dato se usan las siguientes abreviaturas:

Integer =>	int
Double =>	dbl
Long =>	lng
String =>	str
Variant =>	vrn
Fecha =>	fch
Boolean=>	bol

Ejemplos:

Tipo de dato: Boolean

Contenido: falso o verdadero acerca de sí se cargo o no un archivo

El nombre de la variable quedaría así: **bolCargo**