

UNIVERSIDAD TÉCNICA DE COTOPAXI

**CARRERA DE CIENCIAS DE LA INGENIERÍA Y
APLICADAS**

**ESPECIALIDAD INGENIERÍA EN INFORMÁTICA Y
SISTEMAS COMPUTACIONALES**

TESIS DE GRADO

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA MULTIUSUARIO PARA
EL CONTROL, ADMINISTRACIÓN Y GESTIÓN DE LA INFORMACIÓN
PARA LOS ALMACENES “MULTICOMERCIO CENTRO” DE LA CIUDAD
DE LATACUNGA.**

POSTULANTES:

EGDO. LUIS ESTEBAN GUACHO VIZUETE.

EGDO. RAÚL ALFREDO PANCHI HERRERA.

DIRECTOR:

DR. MARCELO BAUTISTA ILLESCAS.

ASESOR:

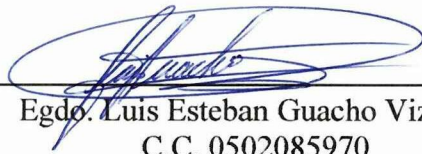
ING. GUIDO SEGOVIA.

LATACUNGA – ECUADOR

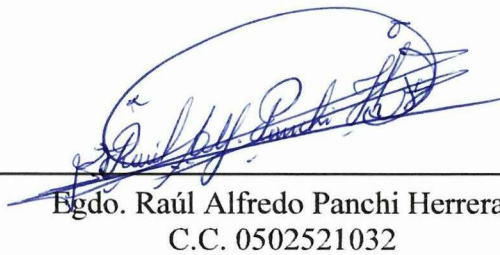
2007

AUTORÍA Y RESPONSABILIDAD

Los autores declaran que la investigación, composición, análisis, desarrollo y propuesta del presente trabajo de tesis son de exclusiva autoría.



Egdo. Luis Esteban Guacho Vizueté.
C.C. 0502085970



Egdo. Raúl Alfredo Panchi Herrera.
C.C. 0502521032

CERTIFICACIÓN

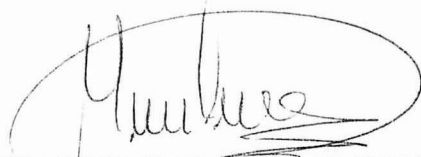
HONORABLE CONSEJO ACADÉMICO DE LA CARRERA EN CIENCIAS DE LA INGENIERÍA Y APLICADAS DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI.

De mi consideración:

Cumpliendo con lo estipulado en el capítulo IV, Art. 9 literal f.) del reglamento del curso preprofesional de la Universidad Técnica de Cotopaxi, informo que los postulantes Guacho Vizúete Luis Esteban y Panchi Herrera Raúl Alfredo, han desarrollado su trabajo de investigación de grado de acuerdo al planteamiento formulado en el plan de tesis con el tema: **“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA MULTIUSUARIO PARA EL CONTROL, ADMINISTRACIÓN Y GESTIÓN DE LA INFORMACIÓN PARA LOS ALMACENES “MULTICOMERCIO CENTRO” DE LA CIUDAD DE LATACUNGA”**, cumpliendo sus objetivos respectivos.

En virtud de lo antes expuesto considero que la presente tesis se encuentra habilitada para presentarse al acto de defensa de tesis.

Latacunga febrero de 2007.



Dr. Marcelo Bautista Illescas.
Director de Tesis

“MULTICOMERCIO CENTRO”



CERTIFICADO

El Dr. Rodrigo Guacho Salazar, Gerente propietario de almacenes “MULTICOMERCIO CENTRO” de la ciudad de Latacunga, tiene a bien certificar que los señores Egdo. Luis Esteban Guacho Vizuite y Egdo. Raúl Alfredo Panchi Herrera, han finalizado el proyecto de tesis con el tema **“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA MULTIUSUARIO PARA EL CONTROL, ADMINISTRACIÓN Y GESTIÓN DE LA INFORMACIÓN PARA LOS ALMACENES “MULTICOMERCIO CENTRO” DE LA CIUDAD DE LATACUNGA”**, el mismo que ha sido entregado, probado en su funcionamiento y está de acuerdo con las especificaciones técnicas, administrativas y de gestión solicitadas.

Motivo por el cual, faculto a los mencionados Señores hacer uso del presente Certificado, como a bien estimaren conveniente.

Latacunga, febrero de 2007.

Dr. Rodrigo Guacho Salazar.

GERENTE PROPIETARIO “MULTICOMERCIO CENTRO”

DEDICATORIA

El presente trabajo lo dedicamos con mucho cariño y afecto a nuestros padres, por brindarnos la vida y en el transcurso de ella cobijarnos con su amor, comprensión, e infinito apoyo, pues de ellos hemos aprendido a luchar hasta en las condiciones más difíciles por alcanzar nuestras metas con el ejemplo de responsabilidad, transparencia y honestidad, porque nos han mostrado que la vida es un collar de pequeños logros y pequeñas perlas de felicidad frente a la inmensidad de la vida.

Esteban, Raúl

AGRADECIMIENTO

Agradecemos a todas las personas que nos apoyaron en todo aspecto durante el desarrollo de la presente tesis, ya sea identificando los problemas, revisando, comentando y corrigiendo el manuscrito, brindándonos su apoyo, su ayuda lógica o técnica, su amistad y compañerismo durante las numerosas veces que se ha requerido colaboración y en especial a:

La Universidad Técnica de Cotopaxi, por ser la cuna de nuestro desarrollo profesional.

Ing. Manuel Ángel León Segovia, por sus consejos y estímulo para seguir creciendo intelectualmente, por su amistad.

Ing. Miguel Patricio Cerda Terán, por su valiosa colaboración y buena voluntad en aclarar inquietudes, así como en sus observaciones críticas durante la redacción del trabajo, por su amistad.

Ing. Guido Orlando Segovia Zambrano, por sus orientaciones críticas y técnicas, por su amistad.

Dr. Marcelo Bautista Illescas, Director de nuestra Tesis, por su paciencia, interés y dedicación al revisar la propuesta y la validación de la tesis; sus ideas permitieron afinar la propuesta y lograr nuestro objetivo, por su amistad.

Dr. Rodrigo Guacho Salazar, por brindarnos su apoyo desinteresado y darnos la apertura total a la información necesaria para culminar nuestro trabajo.

Nuestros padres quienes con su ejemplo de sacrificio y amor, nos infundieron la responsabilidad, ética y moral que tutelan nuestro diario transitar por la vida, guiándonos por el camino del bien y la verdad poniendo la confianza en nosotros, para formarnos como entes útiles a la sociedad, a la Patria y a Dios.

Y en general gracias por la atención y el apoyo de nuestros amigos, compañeros, profesores y profesionales, que directa e indirectamente colaboraron para la culminación de la presente tesis.

A todos ustedes: Gracias... fundamentalmente por su amistad.

Esteban, Raúl.

INDICE GENERAL

CARÁTULA.

AUTORÍA.

CERTIFICADO DEL DIRECTOR DE TESIS.

**CERTIFICADO DE LA EMPRESA COMERCIAL “MULTICOMERCIO
CENTRO”.**

DEDICATORIA.

AGRADECIMIENTO.

ÍNDICE GENERAL.

ÍNDICE DE GRÁFICOS.

ÍNDICE DE TABLAS.

RESUMEN.

SUMMARY.

CERTIFICADO DE INGLÉS.

INTRODUCCIÓN.

CAPÍTULO I

LENGUAJES DE PROGRAMACIÓN Y SISTEMAS OPERATIVOS

1.1. Generalidades acerca de Visual Studio .NET.....	1
1.1.1. Reseña histórica.....	2
1.1.2. Introducción a Visual Studio .NET.....	3
1.1.2.1. Modelo de programación consistente.....	12

1.1.2.2. Modelo de programación sencillo.....	12
1.1.2.3. Integración de lenguajes.....	13
1.1.2.4. Framework .NET.....	14
1.1.2.5. Seguridad avanzada.....	30
1.1.2.6. Requisitos de Hardware.....	39
1.1.3. Visual C# .NET.....	43
1.1.3.1. Generalidades de C#.NET, Lenguaje nativo de .NET.....	43
1.1.3.2. Características y Beneficios de C#.NET.....	46
1.1.3.3. Sencillez.....	47
1.1.3.4. Modernidad.	48
1.1.3.5. Orientación a objetos.....	49
1.1.3.6. Orientación a componentes.....	49
1.1.3.7. Instrucciones seguras.....	50
1.1.3.8. Eficiencia.....	50
1.1.3.9. Introducción a las páginas de Formularios Web Forms.....	51
1.1.3.9.1. Componentes de los Formularios Web Forms.....	52
1.1.3.9.2. Ventajas y Beneficios que otorgan las páginas en Formularios Web Forms.....	55
1.1.3.9.2.1. Implementar una interfaz de usuario Web enriquecida.....	55
1.1.3.9.2.2. Separación entre Cliente / Servidor.....	55
1.1.3.9.2.3. Ejecución Independiente.....	56
1.1.3.9.2.4. Compilaciones con el acceso a los datos.....	56
1.1.3.9.2.5. Compilaciones con la escalabilidad.....	57
1.1.3.9.3. Derivar de la clase Page.....	57

1.1.3.9.4. Elementos de las aplicaciones WEB ASP.NET.....	59
1.1.3.9.4.1. Introducción a ASP.NET.....	59
1.1.3.9.4.2. Funciones de administración de proyectos.....	60
1.1.3.9.4.3. Interfaz de usuario.....	60
1.1.3.9.4.4. Componentes.....	61
1.1.3.9.4.5. Datos.....	61
1.1.3.9.4.6. Seguridad, rendimiento y otras características de infraestructura.....	61
1.1.4. ADO .Net.....	63
1.1.4.1. ADO Net. Características.....	64
1.1.4.2. Interacciones con la base de datos.....	64
1.1.4.3. ADO.NET no depende de conexiones continuamente activas.....	65
1.1.4.4. Los conjuntos de datos son independientes de los orígenes de datos.....	66
1.1.4.5. Componentes de ADO.Net.....	67
1.2. SISTEMAS OPERATIVOS PARA SERVIDORES.....	68
1.2.1. Sistema Operativo Windows XP Profesional.....	68
1.2.1.1. Introducción.....	68
1.2.1.2. Características de Windows XP Profesional.....	69
1.2.2. Sistema Operativo Microsoft Windows 2003 Server.....	71
1.2.2.1. Introducción.....	71
1.2.2.2. Características de Windows 2003 Server.....	72
1.3. BASES DE DATOS.....	74
1.3.1. Concepto de Bases de datos.....	74
1.3.2. RDBMS.....	75

1.3.2.1. Que es un RDBMS.....	75
1.3.3. Concepto de SQL Server.....	76
1.3.3.1. Introducción a SQL Server y a las Bases de Datos Relacionales.....	76
1.3.3.2. Características de SQL Server.....	77
1.3.3.3. SQL Server y el Modelo Cliente / Servidor.....	80
1.3.3.3.1. Ventajas de los clientes.....	81
1.3.3.3.2. Ventajas del Servidor.....	82

CAPITULO II

TRATAMIENTO DE LA INFORMACIÓN DE LA ACTIVIDAD COMERCIAL DE LOS ALMACENES “MULTICOMERCIO CENTRO” DE LA CIUDAD DE LATACUNGA.

2.1. ALMACENES “MULTICOMERCIO CENTRO”.....	83
2.1.1. Datos generales de los Almacenes “Multicomercio Centro”.....	83
2.1.2. Misión.....	84
2.1.3. Visión.....	84
2.1.4. Objetivos.....	84
2.1.5. Política.....	85
2.1.6. Estrategias.....	86
2.1.7. Recursos.....	86
2.1.7.1. Humanos.....	86
2.1.7.2. Infraestructura.....	86
2.1.7.3. Económicos.....	87

2.1.7.4. Tecnológicos.....	87
2.1.7.5. Recursos tecnológicos implementados.....	88
2.2. EL MANEJO DE LA INFORMACIÓN EN LOS ALMACENES	
“MULTICOMERCIO CENTRO”.....	89

CAPÍTULO III

ANÁLISIS DE RESULTADOS.

3.1. Criterio del Propietario – Administrador.....	91
3.2. Criterios de los Clientes.....	97
3.3. Criterios de los Proveedores.....	103
3.4. Criterios de los Vendedores.....	109
3.5. Análisis experiencias similares de otros locales comerciales afines.....	115

CAPÍTULO IV

4.1. Propuesta

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA MULTIUSUARIO PARA EL CONTROL, ADMINISTRACIÓN Y GESTIÓN DE LA INFORMACIÓN PARA LOS ALMACENES “MULTICOMERCIO CENTRO” DE LA CIUDAD DE LATACUNGA.....	116
4.1.1. Modelo de Fases para el desarrollo del software.....	116
4.1.1.1. Análisis.....	116
4.1.1.1.1. Fase de Planificación y Especificación de Requisitos.....	116

4.1.1.1.1.1. Planificación.....	117
4.1.1.1.1.1.1. Propósito.....	117
4.1.1.1.1.1.2. Ámbito del Sistema.....	117
4.1.1.1.1.1.3. Alcance del Sistema.....	118
4.1.1.1.1.1.4. Definición de Usuarios.....	118
4.1.1.1.2. Descripción General.....	119
4.1.1.1.2.1. Funciones del Sistema.....	119
4.1.1.1.3. Requisitos Específicos.....	120
4.1.1.1.3.1. Requisitos Funcionales.....	120
4.1.1.1.3.1.1. Gestión de Clientes.....	120
4.1.1.1.3.1.2. Gestión de Empleados.....	121
4.1.1.1.3.1.3. Gestión de Proveedores.....	121
4.1.1.1.3.1.4. Gestión de Productos y Artículos.....	121
4.1.1.1.3.1.5. Gestión de Facturación.....	122
4.1.1.1.3.2. Requisitos de Rendimiento.....	122
4.1.1.1.3.3. Requisitos Tecnológicos.....	122
4.1.1.1.3.3.1. Requisitos de Hardware.....	122
4.1.1.1.3.3.2. Requisitos de Software.....	123
4.1.1.1.3.4. Requisitos de Desarrollo.....	124
4.1.1.1.3.5. Requisitos de Comunicación.....	124
4.1.1.1.3.6. Atributos.....	124
4.1.1.1.3.6.1. Seguridad.....	124
4.1.1.1.3.6.2. Tipos de Usuarios.....	125

4.1.1.1.3.6.2.1. Administrador del Sistema.....	125
4.1.1.1.3.6.2.2. Empleados.....	125
4.1.1.1.3.6.2.3. Clientes.....	125
4.1.1.1.3.6.2.4. Proveedores.....	126
4.1.1.2. Diseño.....	126
4.1.1.2.1. Diagrama de Casos de Uso.....	127
4.1.1.2.2. Diagrama de Secuencias.....	130
4.1.1.2.3. Diagrama de Clases.....	131
4.1.1.2.4. Diseño Lógico.....	132
4.1.1.2.5. Diseño Físico.....	136
4.1.1.2.6. Diccionario de Datos.....	139
4.1.1.3. Codificación.....	143
4.1.1.3.1. Código Clasificación.....	143
4.1.1.3.2. Código Función.....	144
4.1.1.3.3. Código Secuencia.....	144
4.1.1.3.4. Código Dimito Significativo.....	144
4.1.1.3.5. Código Nematécnico.....	145
4.1.1.4. Pruebas del Sistema.....	146
4.1.1.4.1. Prueba de Datos.....	147
4.1.1.4.2. Prueba de Validaciones.....	148
4.1.1.4.3. Prueba de Seguridades.....	149
4.1.1.5. Fase de Mantenimiento del Sistema.....	152
4.1.2. Elaboración del Manual del Usuario del Sistema.....	153

4.1.3. Elaboración del Manual del Programador del Sistema.....	153
4.1.4. Implementación del Sistema.....	153
4.1.5. Ciclo de vida del Sistema.....	154

CAPÍTULO V

5.1. CONCLUSIONES.....	155
5.2. COMPROBACIÓN DE LA HIPÓTESIS PLANTEADA	156
5.3. RECOMENDACIONES.....	156
5.4. GLOSARIO DE TÉRMINOS.....	158
5.5. BIBLIOGRAFÍA.....	180
5.5.1. Bibliografía Consultada.....	180
5.5.2. Bibliografía Citada.....	180
5.6. ANEXOS.....	181

INDICE DE GRÁFICOS

Figura 1. Componentes de Microsoft .Net.....	4
Figura 2. .Net Framework en contexto.....	19
Figura 3. Elementos aplicaciones Cliente - Servidor.....	25
Figura 4. Código administrado en el servidor.....	28
Figura 5. Estructura de los archivos de formularios Web Forms.....	54
Figura 6. Estructura de la página de formularios Web Forms y la clase Page básica.....	58
Figura 7. Contexto de .NET Framework.....	62
Figura 8. Componentes de ADO.NET.....	67
Figura 9. Encuesta al Administrador de los almacenes “Multicomercio Centro”.....	92
Figura 10. Encuesta al Administrador de los almacenes “Multicomercio Centro”.....	94
Figura 11 - 15. Encuesta al Administrador de los almacenes “Multicomercio Centro” (Ver ANEXO 2 en el CD).	
Figura 16. Encuesta a los clientes de los almacenes “Multicomercio Centro”.....	98
Figura 17. Encuesta a los clientes de los almacenes “Multicomercio Centro”.....	100
Figura 18 - 22. Encuesta a los clientes de los almacenes “Multicomercio Centro” (Ver ANEXO 2 en el CD).	
Figura 23. Encuesta a los proveedores de los almacenes “Multicomercio Centro”.....	104
Figura 24. Encuesta a los proveedores de los almacenes “Multicomercio Centro”.....	106
Figura 25 - 27. Encuesta a los proveedores de los almacenes “Multicomercio Centro” (Ver ANEXO 2 en el CD).	
Figura 28. Encuesta a los vendedores de los almacenes “Multicomercio Centro”.....	110

Figura 29. Encuesta a los vendedores de los almacenes “Multicomercio Centro”.....	112
Figura 30 - 35. Encuesta a los vendedores de los almacenes “Multicomercio Centro” (Ver ANEXO 2 en el CD).	
Figura 36. Diagrama de Casos.....	130
Figura 37. Diagrama de Secuencias.....	131
Figura 38. Diagrama de Clases.....	132
Figura 39. Arquitectura física de tres capas de la aplicación cliente/servidor.....	137
Figura 40. Ingreso del Número de Cédula (Página de ingreso de nuevo cliente).....	148
Figura 41. Errores al ingresar datos de un cliente.....	149
Figura 42. Autenticación ASP.NET (Página principal).....	150
Figura 43. Prueba de Seguridades.- Errores (Encriptación algoritmo Hash (SHA1)).....	151

INDICE DE TABLAS

Tabla 1. Características de las ediciones de Visual Studio .Net.....	6
Tabla 2. Requisitos de software de Visual Studio .Net.....	11
Tabla 3. Requisitos de hardware según Ediciones de Visual Studio .Net.....	40
Tabla 4. Requisitos de hardware Edición Estándar de Visual Studio .Net.....	41
Tabla 5. Encuesta al Administrador de los almacenes “Multicomercio Centro”.....	113
Tabla 6. Encuesta al Administrador de los almacenes “Multicomercio Centro”.....	115
Tabla 7 - 11. Encuesta al Administrador de los almacenes “Multicomercio Centro” (Ver ANEXO 2 en el CD).	
Tabla 12. Encuesta a los Clientes de los almacenes “Multicomercio Centro”.....	128
Tabla 13. Encuesta a los Clientes de los almacenes “Multicomercio Centro”.....	130
Tabla 14 - 18. Encuesta a los Clientes de los almacenes “Multicomercio Centro” (Ver ANEXO 2 en el CD).	
Tabla 19. Encuesta a los Proveedores de los almacenes “Multicomercio Centro”.....	143
Tabla 20. Encuesta a los Proveedores de los almacenes “Multicomercio Centro”.....	145
Tabla 21 - 23. Encuesta a los Proveedores de los almacenes “Multicomercio Centro” (Ver ANEXO 2 en el CD).	
Tabla 24. Encuesta a los Vendedores de los almacenes “Multicomercio Centro”.....	154
Tabla 25. Encuesta a los Vendedores de los almacenes “Multicomercio Centro”.....	156
Tabla 26 - 31. Encuesta a los Vendedores de los almacenes “Multicomercio Centro” (Ver ANEXO 2 en el CD).	
Tabla 32. Población general de los almacenes “Multicomercio Centro”.....	118

Tabla 33. Usuarios del Sistema.....	118
Tabla 34. Gestión de Procesos.....	120
Tabla 35. Prueba de Datos	147
Tabla 36. Prueba de Validaciones	148
Tabla 37. Prueba de Seguridades	150
Tabla 38. Prueba de Seguridades [Encriptación algoritmo Hash (SHA1)].....	152

RESUMEN

Actualmente el grado de desarrollo y adelanto tecnológico que han alcanzado diversos campos como la medicina, la exploración del universo, las comunicaciones, la informática, la electrónica, la robótica, la industria, las empresas comerciales y muchas otras áreas, constituye el baluarte del desarrollo y crecimiento de la humanidad.

Los sistemas informáticos y computacionales al estar inmersos en las nuevas tecnologías de la información y comunicación, están potenciando el desarrollo tecnológico y mejorando la gestión informática de las empresas, en tal virtud el presente trabajo: “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA MULTIUSUARIO PARA EL CONTROL, ADMINISTRACIÓN Y GESTIÓN DE LA INFORMACIÓN PARA LOS ALMACENES “MULTICOMERCIO CENTRO” DE LA CIUDAD DE LATACUNGA”, constituye un aporte para el desarrollo y adelanto comercial de dicha empresa.

La presente tesis contempla el desarrollo e implementación de un software que agilite y facilite la realización de los procesos internos de la casa comercial, permitiendo brindar una mejor atención a los clientes y mantener una buena administración y gestión de la información, la misma ha sido documentada bajo notación estándar y desarrollada con tecnologías de última generación.

La estructura del presente trabajo presenta los conceptos principales del diseño del software, desde el análisis hasta la implementación, partiendo del código fuente, verificando que cumple los requisitos de calidad establecidos en la instalación para poder ser implantado y explotado.

Finalmente el software se ha desarrollado con la utilización del paquete de programación Visual Studio.NET dentro del cuál la programación se ha efectuado en C#.NET. El Análisis y Diseño de la Base de Datos se lo ha efectuado en Rational Rose y Power Designer para finalmente aplicarla en SQL Server como herramienta de Bases de Datos. Todas estas herramientas de desarrollo de software, actualmente tienen el soporte nativo para este tipo de aplicaciones y brindan una interfaz gráfica de usuario que hace fácil familiarizarse con sus componentes.

SUMMARY

At the moment the development grade and technological advance that have reached diverse fields like the medicine, the exploration of the universe, the communications, the computer science, the electronics, the robotics, the industry, the commercial enterprises and many other areas, it constitutes the rampart of the development and the humanity's growth.

The computer and computational systems are involved in the new of the information and communication technologies, they are developed in technological expansion and improving the computer administration of the companies, for this reason this present work: "DESIGN AND IMPLEMENTATION OF A MULTI-USER SYSTEM FOR THE CONTROL, ADMINISTRATION AND MANAGEMENT OF THE INFORMATION FOR THE WAREHOUSES "MULTICOMERCIO CENTER" OF THE LATACUNGA CITY", constitute a contribution for the unfolding and commercial advanced of this company.

The present thesis contemplates the development and implementation of a software that it activates and facilitates the realization of the internal processes of the commercial house, allowing to offer a better attention to the clients and to maintain a good administration and management of the information, the same has been documented down a standard and development with technologies of last generation.

The structure of the present work presents the main concepts of the software design, since the analysis until the implementation, starting of the source code, verifying that it

completes the established requirements of quality in the installation to be able to be implanted and exploited.

Finally the software has been developed with the use of the package of programming Visual Studio .NET. Inside of it, the programming has effected in C#.NET. The Analysis and Design of the Database has made in Rational Rose and Power Designer for finally to apply it in SQL Server like tool of Database. All these tools of software of development, at present has the native support for this type of applications and after a user's graphic interface that makes easy to familiarize with its components.

CERTIFICADO

En calidad de docente del Area de Idiomas de la Universidad Técnica de Cotopaxi, tengo a bien certificar que la traducción al idioma Inglés del RESUMEN DE LA TESIS con el tema: **DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA MULTIUSUARIO PARA EL CONTROL, ADMINISTRACIÓN Y GESTIÓN DE LA INFORMACIÓN PARA LOS ALMACENES “MULTICOMERCIO CENTRO” DE LA CIUDAD DE LATACUNGA**, fue realizada por los señores Egdo. Luis Esteban Guacho Vizuite y Egdo. Raúl Alfredo Panchi Herrera, siendo satisfactoria.

Por tal motivo, faculto a los mencionados Señores hacer uso del presente Certificado, como a bien estimen conveniente.

Latacunga, febrero de 2007.



Lic. Patricia Mena.

DOCENTE DEL AREA DE IDIOMAS DE LA UTC.

INTRODUCCIÓN

El nuevo entorno económico en que vivimos ha dado lugar a que el comercio modifique sus estrategias, planteándose nuevos horizontes y diversos objetivos que le permitan insertarse de manera dinámica en los escenarios multilaterales donde se negocian los lineamientos de política comercial para sus productos.

Sin duda, hoy no existe rubro comercial o actividad profesional que no requiera software o una solución informática. Sin embargo, muchos usuarios instalan un sistema sólo para cumplir requerimientos fiscales o porque la naturaleza de su trabajo (en el caso de los estudios contables, el procesamiento de datos de clientes) así lo requiere, y desconocen las posibilidades potenciales como herramienta de gestión que un software administrativo puede brindarles.

Los sistemas automatizados en los negocios constituyen una de las herramientas más útiles en lo que respecta a la facilitación de los procedimientos que rigen el comercio. Como complemento a la reforma comercial, la automatización es un catalizador de la modernización de las empresas comerciales y un estímulo para un mayor uso de la tecnología de la información de última generación.

Por ello los almacenes “Multicomercio Centro”, de la ciudad de Latacunga, al ser entidades comerciales con objetivos y metas bien establecidas, incursionan en el ámbito tecnológico con el desarrollo del presente sistema informático, que brindará las facilidades necesarias y

requeridas para la fiel satisfacción de sus necesidades, potenciando tanto administrativa, social, comercial y económicamente sus actividades.

En primera instancia se abarcan aspectos de vital importancia como son los lenguajes de programación que serán utilizados como herramientas para el desarrollo del sistema informático, teniendo así un punto de referencia respecto a las tecnologías modernas.

Enseguida se muestra detalladamente las encuestas aplicadas a personas que influyen directamente en los procesos de tratamiento de la información dentro de los almacenes “Multicomercio Centro”, recogiendo sus criterios mediante preguntas formuladas estratégicamente, destacándose la imperiosa necesidad del desarrollo de un software que agilice las operaciones manuales y su correspondiente implementación.

El cuarto capítulo se centra concretamente al diseño e implementación de un sistema multiusuario para el control, administración y gestión de la información para los almacenes “Multicomercio Centro”, donde se detalla toda la información necesaria para la comprensión del mismo mediante una notación estándar en relación al desarrollo de software orientado a objetos.

Finalmente se muestra las conclusiones y recomendaciones, así como se adjunta información adicional para sustentar la veracidad de la presente tesis.

CAPÍTULO

I

**LENGUAJES DE PROGRAMACIÓN Y SISTEMAS
OPERATIVOS**

LENGUAJES DE PROGRAMACIÓN Y SISTEMAS OPERATIVOS

1.1. Generalidades acerca de Visual Studio .NET.

Visual Studio es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones Web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C++, Visual C# y Visual J# utilizan el mismo entorno de desarrollo integrado (IDE), que les permite compartir herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes aprovechan las funciones de .NET Framework, que ofrece acceso a tecnologías clave para simplificar el desarrollo de aplicaciones Web ASP y Servicios Web XML.

Visual Studio .NET brinda funciones de integración para aprovechar aplicaciones existentes en la organización y optimizar los procesos empresariales con clientes y socios. La arquitectura abierta permite a los programadores utilizar cualquier lenguaje orientado a Microsoft .NET Framework y aprovechar los conocimientos de programación actuales, ahorrando así los cursos de reciclaje largos y costosos. Visual Studio .NET está basado en la más reciente plataforma de servidor de Microsoft Windows®, lo que incorpora escalabilidad, confiabilidad y seguridad a las aplicaciones. Se han simplificado la administración y la implementación de las aplicaciones en un ambiente de producción, reduciendo así los costes totales del ciclo.

1.1.1. Reseña histórica.

Desde la aparición de Java, Microsoft ha querido ser su competencia, para ello inicialmente firmó algunos convenios con Sun, para trabajar con Java igual que lo estaban haciendo IBM y Oracle, pero en realidad trató de crear su versión propia de Java, denominada J++, la cual no era completamente compatible con la versión estándar. Además introdujo algunas modificaciones en sus sistemas operativos y navegadores que hacían que el desempeño de aplicaciones desarrolladas en Java fuese más lento de lo que debía.

Pero en el momento en que Java empieza a consolidarse como el lenguaje de Internet, la decisión es más fuerte, hay que lanzar “algo” que realmente le haga contrapeso a Java, y que pueda ser considerado como la competencia de Java en Internet, y a raíz de ello aparece .NET.

Y aquí hay que tratar de revisar qué es .NET, inicialmente se podría pensar que es un lenguaje de programación orientado a objetos, o un conjunto de lenguajes de programación todos ellos orientados a objetos, y se pueden citar C#.NET, C++.NET y VisualBasic.NET; pero a esta última definición le hace falta incluir lo referente al acceso a las bases de datos, es decir ADO.NET, las herramientas para desarrollo en Internet, ASP.NET y el conjunto de facilidades para construir Servicios Web, además del hecho de que todo viene integrado dentro de un entorno completo denominado VisualStudio.NET.

Para lograr elaborar estas herramientas y tener la certeza de que se integrarán bien, se definieron los siguientes elementos:

- Un lenguaje común de ejecución: CLR (Common Language Runtime).
- Un conjunto de tipos de datos básicos: CTS (Common Type System), el cual incluye, además de todos los tipos de datos básicos, las clases Object y String.
- Un CLS (Common Language Specification), que es el conjunto de reglas que especifica lo referente a la implementación de las características de la POO y a otras estructuras sintácticas.
- Un MSIL (MicroSoft Intermediate Language), o lenguaje intermedio común. Este lenguaje es el equivalente al bytecode de Java.
- Un compilador capaz de traducir del MSIL a lenguaje binario, comúnmente denominado JIT (Just In Time).

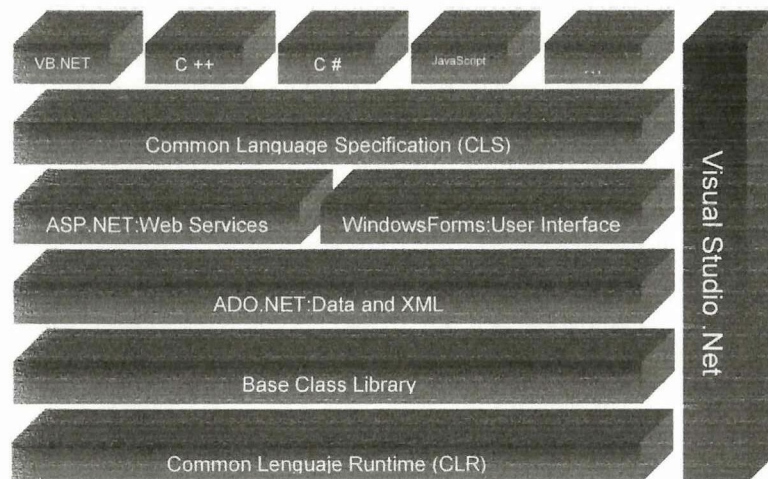
1.1.2. Introducción a Visual Studio .NET.

Cada cierto tiempo se produce una revolución en la tecnología empleada en el desarrollo de aplicaciones. El primer gran cambio que vivimos, fue el paso de la programación en MSDOS a la programación Windows (Win16), más tarde con la evolución de los sistemas operativos, el segundo cambio, las aplicaciones Win32, y ahora estamos viviendo la tercera de las revoluciones, lo que Microsoft denomina .NET Framework o más comúnmente .NET.

Microsoft lanza esta nueva tecnología como respuesta a tecnología Java de Sun. El Framework .NET tiene grandes similitudes con la plataforma Java, por eso todos aquellos que estén familiarizados con Java comprenderán en seguida el funcionamiento de .NET.

Microsoft.NET es el conjunto de nuevas tecnologías en las que Microsoft ha estado trabajando durante los últimos años con el objetivo de obtener una plataforma sencilla y potente para distribuir el software en forma de servicios que puedan ser suministrados remotamente y que puedan comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados. Ésta es la llamada **plataforma .NET**, y a los servicios antes mencionados se les denomina **servicios Web**.

Figura 1. Componentes de Microsoft .Net



Fuente: Ayuda MSDN; de Microsoft Visual Studio. Net, año 2003

El componente principal de .NET, que está en la capa más baja de su modelo de capas, es el *Common language runtime* (CLR), o máquina virtual común. Se trata de un programa, que se puede ejecutar, en principio, en cualquier sistema operativo, y que provee de una serie de servicios que se pueden usar desde diferentes lenguajes de programación.

Los ejecutables CLR, están escritos en un lenguaje denominado **MSIL** (Microsoft intermediate language), similar al Java bytecode; en principio, cualquier programa escrito en MSIL (aunque nadie escribe en MSIL, se supone que lo hacen los compiladores) puede ejecutarse en cualquier sistema operativo donde funcione un CLR; en .NET se usan *ensamblajes* (*assemblies*), que pueden incluir partes de código, datos, códigos de seguridad, y todo lo necesario para convertirlo en código móvil y fiable (en el sentido de que esté firmado por alguien), que se pueda mover por la Internet.

Tabla 1. Características de las ediciones de Visual Studio .Net

Característica	Professional	Enterprise Developer	Enterprise Architect	Academic
Visual Basic .NET	X	X	X	X
Visual C++ .NET	X	X	X	X
Visual C# .NET	X	X	X	X
Visual J# .NET	X	X	X	X
Crear y utilizar servicios Web XML	X	X	X	X
Generar aplicaciones Web	X	X	X	X
Generar aplicaciones para Windows	X	X	X	X
Dispositivos de mano e inalámbricos de destino	X	X	X	X
Equipos Pocket PC y dispositivos con tecnología Windows CE .NET de destino	X	X	X	X
Diseñar tablas y vistas en SQL Server Desktop Engine	X	X	X	X
Diseñar tablas, vistas, procedimientos, desencadenadores, funciones, etc. en SQL Server Desktop Engine, SQL Server, y no en Oracle.		X	X	
Windows 2000 Server Developer		X	X	
SQL Server 2000 Developer		X	X	
Commerce Server 2000 Developer		X	X	
Host Integration Server 2000 Developer		X	X	

Exchange Server 2000 Developer				X	X	
Visual SourceSafe				X	X	
Aplicaciones y servicios Web XML de prueba				X	X	
Aplicaciones de referencia de .NET				X	X	
Ejecución de plantillas de Enterprise Templates				X	X	
Creación de plantillas de Enterprise Templates					X	
BizTalk Server 2000 Developer					X	
Modelado de software basado en Visio					X	
Modelado de bases de datos basado en Visio					X	
Herramientas para el profesorado, incluida la administración de asignaciones y la extracción de código						X
Herramientas para estudiantes, incluidos asistentes para aplicaciones y administración de asignaciones						X
Documentación académica específica y ejemplos de código para profesores y estudiantes						X

Fuente: Ayuda MSDN; de Microsoft Visual Studio. Net, año 2003¹

¹ ms-help://MS.MSDNQTR.2003FEB.3082/vsimtro7/html/vxort/VisualStudioNET/SystemRequirements.htm

Instalación y Configuración de Visual Studio .NET

Cuando se instala Visual Studio .NET, se pueden elegir diversas opciones de instalación y configuración, como se detalla a continuación:

Instalaciones simultáneas de Visual Studio .NET

Visual Studio admite la instalación de las versiones 2002 y 2003 en el mismo equipo, lo que permite evaluar la versión más reciente de Visual Studio y actualizar a esa versión de forma gradual.

Es necesario tener en cuenta las siguientes condiciones:

- Si abre una solución creada con Visual Studio .NET 2002 en Visual Studio .NET 2003 y guarda la solución, ya no podrá abrir la solución en Visual Studio .NET 2002.
- Visual Studio .NET 2002 se distribuye con la versión 1.0 de Microsoft .NET Framework SDK; Visual Studio .NET 2003 con la versión 1.1. Si tiene instaladas las dos versiones de .NET Framework, Visual Studio .NET 2003 le permite elegir la versión que admiten los proyectos. Los proyectos de Visual Basic, Visual C# y Visual J# utilizan una nueva propiedad denominada Motores en tiempo de ejecución compatibles. Visual

C++ ofrece una forma de especificar manualmente los motores en tiempo de ejecución compatibles.

- Es posible copiar determinadas configuraciones del cuadro de diálogo Opciones de Visual Studio .NET 2002 a Visual Studio .NET 2003.
- Los cambios en la forma de controlar los identificadores de programa (ProgID) en el modelo de automatización de Visual Studio .NET 2003 pueden producir errores con los componentes escritos en Visual Studio .NET 2002. Además, para invocar la creación de nuevas instancias de Visual Studio .NET IDE ahora es necesario un ProgID dependiente de la versión.

Permisos de usuario

Para instalar el producto, es necesario tener permisos de administrador en el equipo. Sin embargo, los permisos de administrador no son necesarios para utilizar Visual Studio después de la instalación. La mayoría de las herramientas sólo exigen los permisos básicos que proporciona el grupo Usuarios. Visual Studio tiene sus propios grupos de usuarios, como Usuarios del depurador o Programadores de VS, para determinadas herramientas. Por ejemplo, para crear proyectos Web en el equipo local, es necesario ser miembro del grupo Programadores de VS.

Opciones de configuración avanzadas

La configuración de Visual Studio .NET ofrece dos modos de instalación especializados, Administrador y Componentes remotos.

Instalación en modo Administrador

Este modo de instalación permite a los administradores de red efectuar implementaciones en los equipos cliente de forma silenciosa. La instalación en modo administrador permite crear archivos de instalación personalizados para Visual Studio .NET Prerrequisitos y para la instalación principal de Visual Studio .NET.

Instalación de componentes remotos

Este modo permite instalar los componentes seleccionados, como componentes de depuración y de análisis, en equipos remotos.

Requisitos de software de Visual Studio .NET

Cierto tipo de proyectos y funciones de Visual Studio precisan de la instalación de componentes de software específicos, que aparecen como opcionales durante la configuración, antes de poder utilizar dicho proyecto o función. Algunos de estos componentes deben instalarse en

el equipo de desarrollo, mientras que otros pueden instalarse en un equipo remoto.

En la Tabla 2, se muestra una lista de componentes que deben estar instalados en los distintos sistemas operativos para efectuar las tareas especificadas.

Tabla 2. Requisitos de software de Visual Studio .Net

Desea	Windows 2000	Windows XP, Windows Server 2003 o posterior	Windows NT4,¹ Windows 98, Windows Me, Windows XP Home
Desarrollar aplicaciones Web ASP y servicios Web XML	Servicios de Internet Information Server (IIS)	IIS	No admitido
Compilar código relacionado con Microsoft Windows Message Queuing (MSMQ)	Servicios de Message Queue Server	Servicios de Message Queue Server	No admitido
Depurar código en equipos remotos	Depurador remoto de Visual Studio	Depurador remoto de Visual Studio	Depurador remoto de Visual Studio
Usar el control de código fuente para las versiones de procedimientos almacenados	Versiones de procedimientos almacenados de Visual Studio 6.0 Visual SourceSafe Microsoft SQL Server	Versiones de procedimientos almacenados de Visual Studio 6.0 Visual SourceSafe Microsoft SQL Server	No admitido

Fuente: Ayuda MSDN; de Microsoft Visual Studio. Net, año 2003

¹ Windows NT 4.0 sólo admite la instalación de Visual Studio Remote Debugger y Visual Studio Analyzer Client.

Visual Basic, Visual C++, Visual C#, y Visual J# no se pueden instalar con Windows NT 4.0

1.1.2.1. Modelo de programación consistente.

A todos los servicios y facilidades ofrecidos por el CLR se accede de la misma forma: a través de un modelo de programación orientado a objetos. Esto es una diferencia importante respecto al modo de acceso a los servicios ofrecidos por los algunos sistemas operativos actuales (por ejemplo, los de la familia Windows), en los que a algunos servicios se les accede a través de llamadas a funciones globales definidas en DLLs y a otros a través de objetos (objetos COM en el caso de la familia Windows)

1.1.2.2. Modelo de programación sencillo.

Con el CLR desaparecen muchos elementos complejos incluidos en los sistemas operativos actuales (registro de Windows, GUIDs, HRESULTS, IUnknown, etc.) El CLR no es que abstraiga al programador de estos conceptos, sino que son conceptos que no existen en la plataforma .NET.

Eliminación del “infierno de las DLLs”: En la plataforma .NET desaparece el problema conocido como “infierno de las DLLs” que se da en los sistemas operativos actuales de la familia Windows, problema que consiste en que al sustituirse versiones viejas de DLLs compartidas por versiones nuevas puede que aplicaciones que fueron diseñadas para ser ejecutadas usando las viejas dejen de funcionar si las nuevas no son 100% compatibles con las anteriores. En la plataforma .NET las versiones nuevas de las DLLs pueden coexistir con las viejas, de modo que las aplicaciones diseñadas para ejecutarse usando las viejas podrán seguir usándolas tras instalación de las nuevas. Esto, obviamente, simplifica mucho la instalación y desinstalación de software.

1.1.2.3. Integración de lenguajes.

Desde cualquier lenguaje para el que exista un compilador que genere código para la plataforma .NET es posible utilizar código generado para la misma usando cualquier otro lenguaje tal y como si de código escrito usando el primero se tratase. Microsoft ha desarrollado un compilador de C# que genera código de este tipo, así como versiones de sus compiladores de Visual Basic (Visual Basic.NET) y C++ (C++ con extensiones gestionadas) que también lo generan y una versión del intérprete de JScript (JScript.NET) que

puede interpretarlo. La integración de lenguajes consiste en que es posible escribir una clase en C# que herede de otra escrita en Visual Basic.NET que, a su vez, herede de otra escrita en C++ con extensiones gestionadas.

1.1.2.4. Framework .NET.

Microsoft lanza esta nueva tecnología como respuesta a tecnología Java de Sun. El Framework .NET tiene grandes similitudes con la plataforma Java, por eso todos aquellos que estén familiarizados con Java comprenderán en seguida el funcionamiento de .NET.

Centrándonos en el concepto Framework .NET, como el término en inglés dice (Framework = Armazón) es un marco en donde nuestras aplicaciones correrán. Nuestras aplicaciones ya no corren directamente bajo el sistema operativo si no que corren bajo este armazón o marco.

Elementos principales de .NET Framework:

- CLR (Common Language Runtime)
- El conjunto de clases del .NET Framework
- ASP.NET
- Los servicios Web

- Remoting
- Windows Forms

El **CLR** es el motor de ejecución de las aplicaciones .NET, lo que en Java sería la máquina virtual de Java, este motor se encarga de ejecutar todo el código .NET para ello a de ser en dicho lenguaje. El CLR es el encargado de convertir este lenguaje intermedio en lenguaje máquina del procesador, esto normalmente se hace en tiempo real por un compilador JIT (Just-In-Time) que lleva incorporado el CLR.

El **conjunto de clases del .NET Framework** es la piedra angular de cualquier desarrollador de .NET, es un rico conjunto de clases, interfaces, tipos que simplifican y optimizan el desarrollo de aplicaciones .NET además de proporcionar acceso a la funcionalidad del sistema. Como desarrolladores el dominio de este conjunto de clases es vital para un buen desarrollo en .NET.

ASP.NET es la parte del .NET Framework dedicada al desarrollo web. A través del servidor web (IIS) nuestras aplicaciones ASP.NET se ejecutarán bajo el CLR y podremos usar el conjunto de clases del .NET Framework para desarrollarlas, obteniendo así una versatilidad y una potencia nunca antes conseguida en las aplicaciones ASP.

También son destacables los **servicios web**, que nos permitirán comunicarnos a través de Internet entre diferentes ordenadores, incluso entre distintos sistemas. Así como **.NET Remoting** que nos permite tener objetos en máquinas remotas e invocarlos desde otras máquinas.

Y las **Windows Forms**, parte del .NET Framework que permite crear aplicaciones en el más clásico de los sentidos, es decir Formularios que presentan un entorno Windows.

Información general acerca de .NET Framework.

.NET Framework es una nueva plataforma informática que simplifica el desarrollo de aplicaciones en un entorno altamente distribuido como es Internet. El diseño de .NET Framework está enfocado a cumplir los objetivos siguientes:

- Proporcionar un entorno coherente de programación orientada a objetos, en el que el código de los objetos se pueda almacenar y ejecutar de forma local, ejecutar de forma local pero distribuida en Internet o ejecutar de forma remota.

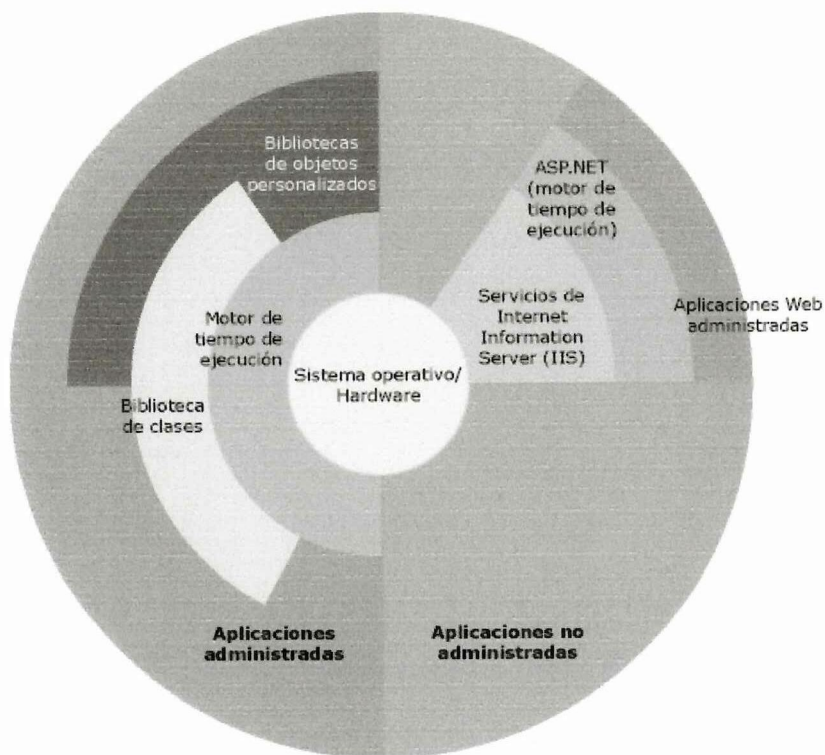
- Proporcionar un entorno de ejecución de código que reduzca lo máximo posible la implementación de software y los conflictos de versiones.
- Ofrecer un entorno de ejecución de código que garantice la ejecución segura del mismo, incluso del creado por terceras personas desconocidas o que no son de plena confianza.
- Proporcionar un entorno de ejecución de código que elimine los problemas de rendimiento de los entornos en los que se utilizan secuencias de comandos o intérpretes de comandos.
- Ofrecer al programador una experiencia coherente entre tipos de aplicaciones muy diferentes, como las basadas en Windows o en el Web.
- Basar toda la comunicación en estándares del sector para asegurar que el código de .NET Framework se puede integrar con otros tipos de código.

.NET Framework contiene dos componentes principales: Common Language Runtime y la biblioteca de clases de .NET Framework. **Common Language Runtime** es el fundamento de la tecnología. El motor de tiempo de ejecución se puede considerar como un agente que administra el código en tiempo de ejecución y proporciona

servicios centrales, como la administración de memoria, la administración de subprocesos y la interacción remota, al tiempo que aplica una seguridad estricta a los tipos y otras formas de especificación del código que garantizan su seguridad y solidez. De hecho, el concepto de administración de código es un principio básico del motor de tiempo de ejecución. El código destinado al motor de tiempo de ejecución se denomina *código administrado*, a diferencia del resto de código, que se conoce como *código no administrado*. **La biblioteca de clases**, el otro componente principal de .NET Framework, es una completa colección orientada a objetos de tipos reutilizables que se pueden emplear para desarrollar aplicaciones que abarcan desde las tradicionales herramientas de interfaz gráfica de usuario (GUI) o de línea de comandos hasta las aplicaciones basadas en las innovaciones más recientes proporcionadas por ASP.NET, como los formularios Web Forms y los servicios Web XML.

En la Figura 2, se muestra la relación de Common Language Runtime y la biblioteca de clases con las aplicaciones y el sistema en su conjunto. También se representa cómo funciona el código administrado dentro de una arquitectura mayor.

Figura 2.. .Net Framework en contexto



Fuente: Ayuda MSDN; de Microsoft Visual Studio. Net, año 2003

Características de Common Language Runtime.

Common Language Runtime administra la memoria, ejecución de subprocesos, ejecución de código, comprobación de la seguridad del código, compilación y demás servicios del sistema. Estas características son intrínsecas del código administrado que se ejecuta en Common Language Runtime.

Con respecto a la seguridad, los componentes administrados reciben grados de confianza diferentes, en función de una serie de factores entre los que se incluye su origen (como Internet, red empresarial o equipo local). Esto significa que un componente administrado puede ser capaz o no de realizar operaciones de acceso a archivos, operaciones de acceso al Registro y otras funciones delicadas, incluso si se está utilizando en la misma aplicación activa.

El motor de tiempo de ejecución impone seguridad en el acceso al código. Por ejemplo, los usuarios pueden confiar en que un archivo ejecutable incrustado en una página Web puede reproducir una animación en la pantalla o entonar una canción, pero no puede tener acceso a sus datos personales, sistema de archivos o red. Por ello, las características de seguridad del motor de tiempo de ejecución permiten que el software legítimo implementado en Internet sea excepcionalmente variado.

Además, el motor de tiempo de ejecución impone la solidez del código mediante la implementación de una infraestructura estricta de comprobación de tipos y código denominada CTS (Common Type System, Sistema de tipos común). CTS garantiza que todo el código administrado es autodescriptivo. Los diferentes compiladores de lenguajes de Microsoft y de terceros generan código administrado

que se ajusta a CTS. Esto significa que el código administrado puede usar otros tipos e instancias administrados, al tiempo que se aplica inflexiblemente la fidelidad y seguridad de los tipos.

Además, el entorno administrado del motor de tiempo de ejecución elimina muchos problemas de software comunes. Por ejemplo, el motor de tiempo de ejecución controla automáticamente la disposición de los objetos, administra las referencias a éstos y los libera cuando ya no se utilizan. Esta administración automática de la memoria soluciona los dos errores más comunes de las aplicaciones: la pérdida de memoria y las referencias no válidas a la memoria.

Además, el motor de tiempo de ejecución aumenta la productividad del programador. Por ejemplo, los programadores pueden crear aplicaciones en el lenguaje que prefieran y seguir sacando todo el provecho del motor de tiempo de ejecución, la biblioteca de clases y los componentes escritos en otros lenguajes por otros colegas. El proveedor de un compilador puede elegir destinarlo al motor de tiempo de ejecución. Los compiladores de lenguajes que se destinan a .NET Framework hacen que las características de .NET Framework estén disponibles para el código existente escrito en dicho lenguaje, lo que facilita enormemente el proceso de migración de las aplicaciones existentes.

Aunque el motor de tiempo de ejecución está diseñado para el software del futuro, también es compatible con el software actual y el software antiguo. La interoperabilidad entre el código administrado y no administrado permite que los programadores continúen utilizando los componentes COM y las DLL que necesiten.

El motor de tiempo de ejecución está diseñado para mejorar el rendimiento. Aunque Common Language Runtime proporciona muchos servicios estándar de motor de tiempo de ejecución, el código administrado nunca se interpreta. Una característica denominada compilación JIT (Just-In-Time) permite ejecutar todo el código administrado en el lenguaje máquina nativo del sistema en el que se ejecuta. Mientras tanto, el administrador de memoria evita que la misma se pueda fragmentar y aumenta la zona de referencia de la memoria para mejorar aún más el rendimiento.

Por último, el motor de tiempo de ejecución se puede alojar en aplicaciones de servidor de gran rendimiento, como Microsoft® SQL Server™ e IIS (Servicios de Internet Information Server). Esta infraestructura permite utilizar código administrado para escribir lógica empresarial, al tiempo que se disfruta del superior

rendimiento de los mejores servidores empresariales del sector que puedan alojar el motor de tiempo de ejecución².

Biblioteca de clases de .NET Framework.

La biblioteca de clases de .NET Framework es una colección de tipos reutilizables que se integran estrechamente con Common Language Runtime. La biblioteca de clases está orientada a objetos, lo que proporciona tipos de los que su propio código administrado puede derivar funciones. Esto ocasiona que los tipos de .NET Framework sean sencillos de utilizar y reduce el tiempo asociado con el aprendizaje de las nuevas características de .NET Framework.

Además, los componentes de terceros se pueden integrar sin dificultades con las clases de .NET Framework.

Por ejemplo, las clases de colección de .NET Framework implementan un conjunto de interfaces que puede usar para desarrollar sus propias clases de colección. Éstas se combinarán fácilmente con las clases de .NET Framework.

Como en cualquier biblioteca de clases orientada a objetos, los tipos de .NET Framework permiten realizar diversas tareas de

² ms-help://MS.MSDNQTR.2003FEB.3082/cpguide/html/cpovrintroductiontonetframeworksdk.htm

programación comunes, como son la administración de cadenas, recopilación de datos, conectividad de bases de datos y acceso a archivos. Además de estas tareas habituales, la biblioteca de clases incluye tipos adecuados para diversos escenarios de desarrollo especializados. Por ejemplo, puede utilizar .NET Framework para desarrollar los siguientes tipos de aplicaciones y servicios:

- Aplicaciones de consola
- Aplicaciones GUI de Windows (Windows Forms)
- Aplicaciones de ASP.NET
- Servicios Web XML
- Servicios de Windows

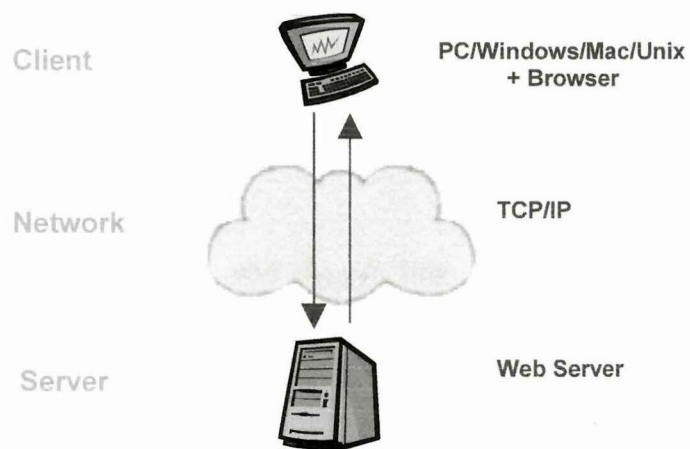
Desarrollo de aplicaciones cliente.

Las aplicaciones cliente constituyen lo más parecido a una aplicación de estilo tradicional en la programación basada en Windows. En este tipo de aplicaciones se muestran ventanas o formularios en el escritorio, lo que permite al usuario realizar una tarea. Entre las aplicaciones cliente se incluyen los procesadores de texto y las hojas de cálculo, además de aplicaciones empresariales, como herramientas de entrada de datos, de informes, etc. En las aplicaciones cliente se suelen emplear ventanas, menús, botones y

otros elementos de la interfaz gráfica de usuario, y suelen tener acceso a recursos locales como el sistema de archivos y a dispositivos periféricos como las impresoras.

Otro tipo de aplicación cliente es el tradicional control ActiveX (reemplazado ahora por el control de Windows Forms) implementado en Internet como una página Web. Esta aplicación es muy parecida a otras aplicaciones cliente: se ejecuta de forma nativa, tiene acceso a los recursos locales e incluye elementos gráficos.

Figura 3. Elementos aplicaciones Cliente - Servidor



Fuente: Internet: <http://www.msn.com/default.asp>

En el pasado, los programadores creaban esas aplicaciones mediante C o C++ en combinación con MFC (Microsoft Foundation Classes) o con un entorno RAD (Rapid Application Development, desarrollo

rápido de aplicaciones) como Microsoft® Visual Basic®. En .NET Framework se incorporan aspectos de estos productos, que siguen existiendo, en un único entorno de desarrollo coherente que simplifica de forma espectacular el desarrollo de las aplicaciones cliente.

Las clases de Windows Forms contenidas en .NET Framework están diseñadas para utilizarse en el desarrollo de GUI (Graphic User Interface). Puede crear ventanas, botones, menús, barras de herramientas y demás elementos de pantalla fácilmente con la flexibilidad requerida para adaptarse a la evolución de las necesidades de su empresa.

Por ejemplo, .NET Framework proporciona propiedades simples para ajustar los atributos visuales asociados con los formularios. En determinadas circunstancias, el sistema operativo subyacente no permite cambiar estos atributos directamente y, entonces, .NET Framework vuelve a crear los formularios de forma automática. Ésta es una de las múltiples maneras en que .NET Framework integra la interfaz del programador, con lo que la creación de código resulta más sencilla y más coherente.

A diferencia de los controles ActiveX, los controles de Windows Forms tienen acceso con una confianza parcial al equipo de un usuario. Esto significa que el código binario o que se ejecuta de forma nativa puede tener acceso a algunos de los recursos del sistema del usuario (como elementos de la GUI y acceso limitado a los archivos) sin tener acceso ni comprometer los demás recursos. Debido a la seguridad de acceso a código, muchas aplicaciones que antes era necesario instalar en el sistema de un usuario, ahora se pueden implementar con seguridad a través del Web. Las aplicaciones pueden implementar las características de una aplicación local a la vez que se implementan como una página Web.

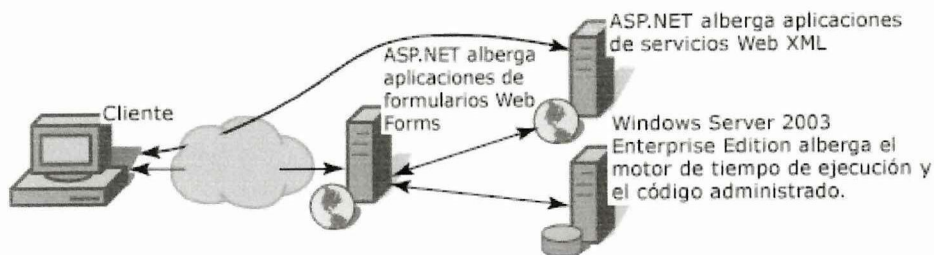
Desarrollo de aplicaciones de servidor.

Las aplicaciones de servidor en entornos administrados se implementan mediante hosts de motor de tiempo de ejecución. Las aplicaciones no administradas alojan Common Language Runtime, que permite al código administrado personalizado controlar el comportamiento del servidor. Este modelo proporciona todas las características de Common Language Runtime y la biblioteca de clases, además de obtener el rendimiento y la escalabilidad del servidor host.

En la ilustración siguiente se muestra un esquema de red básico donde se ejecuta código administrado en diferentes entornos de servidor. Los servidores como IIS y SQL Server pueden realizar operaciones estándar mientras la lógica de la aplicación se ejecuta en el código administrado.

Código administrado en el servidor.

Figura 4. Código administrado en el servidor



Fuente: Ayuda MSDN; de Microsoft Visual Studio. Net, año 2003

ASP.NET es el entorno host que permite a los programadores utilizar .NET Framework con las aplicaciones para el Web. Sin embargo, ASP.NET no es sólo un host de motor de tiempo de ejecución: se trata de una completa arquitectura para el desarrollo de sitios Web y objetos distribuidos en Internet mediante código administrado. Los formularios Web Forms y los servicios Web XML utilizan IIS y ASP.NET como mecanismos de publicación de las aplicaciones y ambos disponen de una colección de clases compatibles en .NET Framework.

Los servicios Web XML, que constituyen una evolución importante de la tecnología basada en el Web, son componentes distribuidos de aplicaciones de servidor similares a los sitios Web comunes. Sin embargo, a diferencia de las aplicaciones basadas en el Web, los componentes de servicios Web XML no tienen interfaz de usuario y no están orientados a exploradores como Internet Explorer y Netscape Navigator. En su lugar, los servicios Web XML consta de componentes de software reutilizables diseñados para que los utilicen otras aplicaciones, como aplicaciones cliente tradicionales, aplicaciones basadas en el Web o, incluso, otros servicios Web XML. Como resultado, la tecnología de servicios Web XML está desplazando rápidamente el desarrollo y la implementación de aplicaciones hacia el entorno altamente distribuido de Internet.

Si ha utilizado versiones anteriores de la tecnología ASP, apreciará de inmediato las mejoras que ofrecen ASP.NET y Web Forms. Por ejemplo, puede desarrollar páginas de Web Forms en cualquier lenguaje compatible con .NET Framework. Además, ya no es necesario que el código comparta el mismo archivo con el texto HTTP (aunque puede seguir haciéndolo, si lo prefiere). Las páginas de Web Forms se ejecutan en lenguaje máquina nativo porque, al igual que todas las aplicaciones administradas, sacan todo el provecho del motor de tiempo de ejecución.

1.1.2.5. Seguridad avanzada.

El CLR proporciona mecanismos para restringir la ejecución de ciertos códigos o los permisos asignados a los mismos según su procedencia o el usuario que los ejecute. Es decir, puede no darse el mismo nivel de confianza a código procedente de Internet que a código instalado localmente o procedente de una red local; puede no darse los mismos permisos a código procedente de un determinado fabricante que a código de otro; y puede no darse los mismos permisos a un mismo código según el usuario que lo esté ejecutando o según el rol que éste desempeñe. Esto permite asegurar al administrador de un sistema que el código que se esté ejecutando no pueda poner en peligro la integridad de sus archivos, la del registro de Windows, etc.

La seguridad, en el contexto de una aplicación ASP.NET (Active Server Page), implica la consecución de tres operaciones fundamentales, que se ejecutan a lo largo del ciclo vital de la aplicación Web ASP.NET segura, siendo mencionadas operaciones las siguientes:

- **Autenticación** Es el proceso de validación de la identidad de un usuario para permitir o denegar una solicitud. Generalmente consiste en aceptar el nombre de usuario y su contraseña y

validarlos posteriormente, dichos datos de una base de datos seguridad.

- **Autorización** En este proceso se comprueba que los usuarios con identidad válida tengan acceso solamente a aquellos recursos a los que tienen derechos o permisos de acceso. Es decir que la autorización es una simple revisión que se ejecuta en todas las fases del procesamiento de solicitudes en el servidor Web. Garantizando de esa forma que únicamente se acceda a los recursos permitidos.
- **Personalización** Este proceso permite que una aplicación asuma la identidad del invocador y que, en su nombre, realice solicitudes a otros recursos. El acceso se concede o deniega en función a la identidad que se adopte. Si esta identidad dispone de permisos para un recurso, la aplicación que la adopta también dispondrá de los mismos permisos de acceso.

Se debe indicar que en las aplicaciones ASP.NET, el servidor Web base es Servicios de Internet Information Server (IIS), el cual proporciona las opciones de seguridad correspondientes, las cuales se detallan a continuación:

SEGURIDAD IIS

La protección de una aplicación Web consta de diferentes aspectos. La mejor forma de empezar a proteger una aplicación es con la ayuda de los métodos de seguridad que proporciona el servidor Web en el que se encuentra la aplicación. El servidor IIS es compatible con la autenticación y autorización de solicitudes de usuario.

Autenticación El servidor IIS es compatible, de forma predeterminada, con la autenticación de solicitudes cliente de contenidos Web almacenados en un sitio Web IIS. Por medio de un servidor IIS se pueden implementar tres tipos diferentes de autenticación:

- **Autenticación anónima.** Permite a todos los usuarios navegar por el sitio Web sin tener que introducir un nombre de usuario y una contraseña. El servidor IIS, por medio de la cuenta IUSR_machinename, personaliza el acceso a los recursos del sitio Web.
- **Autenticación básica.** Exige a los usuarios introducir una combinación de nombre de usuario y contraseña para acceder al sitio Web. El inconveniente de éste método es que la contraseña se envía por la red sin ningún tipo de codificación, por lo que los usuarios no

autorizados pueden conseguir la información de la misma si acceden a los paquetes de la red.

- **Autenticación de Windows integrada.** Exige que los usuarios sean usuarios válidos de Windows además de cumplir con la autenticación básica. En este modo IIS verifica el nombre de usuario y la contraseña por medio de un Controlador de dominio de Windows. Únicamente se concede el acceso al sitio Web si el Controlador de dominio valida el nombre de usuario y la contraseña.
- **Autenticación de compendio.** Similar a la autenticación básica. Sin embargo, este modelo recurre a una forma distinta de transmisión de las credenciales de autenticación. Envía un valor hash por la red en lugar de remitir la contraseña, valor que no se puede decodificar y, por lo tanto, tampoco se puede descifrar del texto original.

Autorización Se puede configurar el servidor IIS para controlar los recursos a los que pueden acceder los usuarios. Los permisos de acceso se controlan mediante el marcado de las operaciones permitidas en el sitio Web. A continuación se describen los distintos niveles de permisos:

- **Lectura.** Permite a los usuarios recuperar y modificar el contenido almacenado en el directorio virtual. Es el permiso que se asigna a la mayor parte de los directorios virtuales.

- **Escritura.** Permite a los usuarios recuperar y modificar el contenido almacenado en el directorio virtual. Si un sitio Web se abre para recibir contenidos a través del protocolo HTTP, el directorio virtual utilizado para almacenar los archivos recibidos debe tener permiso de escritura. Un ejemplo típico sería el de un directorio virtual que almacena archivos cargados como archivos adjuntos de correos electrónicos.
- **Acceso de origen del script.** Permite a los usuarios ver el código fuente de cualquier programa en el lado del servidor.
- **Exploración de directorios.** Permite a los usuarios ver el contenido completo del directorio virtual. Es similar al hecho de ver un directorio de FTP.
- **Registro de visitas.** Registra el número de usuarios que visita un sitio así como información relacionada con otros aspectos como la dirección IP del cliente o los recursos que haya solicitado.
- **Índice.** Utiliza Microsoft Index Server para indexar el directorio virtual. Los contenidos del directorio se pueden recuperar en un resultado de búsqueda por medio de Index Server.

Además de los niveles de permiso IIS, los permisos NTFS se pueden utilizar para proteger los archivos y directorios de un servidor Web. Seguidamente se detallan los distintos permisos de acceso que puede

asignar a usuarios y a grupos con respecto al uso de archivos y directorios de un servidor:

- **Control total.** Permite a los usuarios controlar completamente los archivos y/o directorios.
- **Modificar.** Permite a los usuarios modificar los contenidos de archivos y/o directorios, aunque no puedan eliminarlos.
- **Lectura y Ejecución.** Los usuarios pueden leer el contenido de los archivos y/o directorios existentes y ejecutar cualquier aplicación almacenada en dicha carpeta. Sin embargo, no pueden modificar los contenidos de los archivos ni de los directorios.
- **Lista de contenidos de la carpeta.** Los usuarios pueden ver los contenidos de la carpeta pero no pueden leer ni modificar los contenidos de ninguno de los archivos de la misma.
- **Escritura.** Permite que los usuarios accedan a los archivos y/o directorios.
- **Sin acceso.** No permite que los usuarios accedan a los archivos y/o directorios.

AUTENTICACIÓN EN APLICACIONES WEB

Existen diversas formas de autenticar el acceso de usuarios a aplicaciones Web. En aplicaciones de Intranet, se puede utilizar el

modelo de autenticación que Windows integra para validar el acceso de usuarios e implantar el control de acceso. Sin embargo, en la mayoría de aplicaciones de Internet, no se puede utilizar la autenticación de Windows, ya que existen diversas restricciones como se explican seguidamente:

- **Número de cuentas de usuario.** Aunque el directorio Activo de Windows puede escalar un elevado número de cuentas de usuario, la gestión de todas las cuentas de aplicaciones de Internet (que pueden llegar a ser millones) se convierte en un auténtico reto. Por esta razón, los programadores y administradores Web prefieren utilizar un mecanismo de autenticación basado en bases de datos, como bases de datos SQL.
- **Licencias.** Si hubiera que autenticar millones de usuarios en función de una base de datos de Directorio Activo, el sitio Web tendría que proporcionar licencias de uso a todos ellos, lo que sería casi imposible.

En el modelo ASP clásico, los aspectos de autenticación eran dirigidos por implementaciones de seguridad dependientes de cookies o IP cliente. Esto significaba tener que escribir una gran cantidad de código, una sobrecarga innecesaria para los programadores. Sin embargo, en la

implementación de aspectos de seguridad en aplicaciones de Windows, el desarrollo intenta aprovechar el máximo de servicios que ofrece el sistema operativo. En cambio con ASP .NET, ya no es necesario escribir tanto código. Los programadores se sirven de Microsoft .NET Framework para aplicar los aspectos de seguridad y solamente tienen que preocuparse por solucionar los problemas de negocios y de implementar las pertinentes funciones en el sitio Web.

Opciones de autenticación ASP.NET

Una aplicación Web de ASP.NET puede utilizar los siguientes tipos de seguridad:

- **Windows.** La aplicación se protege por medio de la Autenticación de Windows Integrada, método en el que solamente se concede el acceso a la aplicación Web a los usuarios que pueden verificar sus credenciales de Windows. Las credenciales se verifican en función de la base de datos de autenticación de Windows (SAM) o del Directorio Activo.
- **Pasaporte.** La aplicación se protege por medio de la autenticación Microsoft Passport, tecnología diseñada por Microsoft para ser utilizada en la Web.

- **Formularios.** La aplicación se protege por medio de un modelo de autenticación personalizado compatible con los cookies.
- **Ninguno.** La aplicación no se protege y el acceso a la misma no requiere autenticación.

Autenticación basada en formularios

ASP.NET incluye una opción predeterminada denominada autenticación basada en formularios, que se puede utilizar para implementar lógica personalizada para autenticar a los usuarios y controladores de autenticación sin necesidad de preocuparse de la gestión de sesión por medio de cookies. En este tipo de autenticación, cuando un usuario no es autenticado, se le dirige automáticamente a la página de registro. Entre las ventajas de este modelo de autenticación, podemos destacar las siguientes:

- Los programadores pueden configurar la autenticación basada en formularios para distintas partes de sitio Web ya que el archivo Web.config es un documento XML jerárquico.
- Los administradores y programadores pueden modificar el esquema de autenticación de forma rápida y sencilla en el archivo Web.config.

- La administración es centralizada ya que todas las entradas de autenticación se encuentran en el mismo punto, el archivo Web.config.

Seguridad de acceso al código para restringir el uso de ensamblados

En algunas aplicaciones, es posible que se desee restringirse el acceso a clases. Un método para ello es hacer que las clases sean privadas o internas, de tal forma que se garantice que no se pueda llamar a ellas desde fuera del ensamblado en el que residen. Sin embargo, la mayoría de las aplicaciones están separadas en varios ensamblados por numerosas razones; por ejemplo, la repartición física y lógica de las aplicaciones y el uso compartido del código entre componentes. En estos casos, será necesario un mecanismo para restringir las llamadas entre ensamblados³.

1.1.2.6. Requisitos de Hardware de Visual Studio .NET.

El equipo en el que instale la edición de Visual Studio .NET debe cumplir los requisitos siguientes.

Requisitos según Ediciones de Visual Studio .NET

³ Consulta Bibliográfica; Anaya Multimedia; MRIDULA PARIHAR; La Biblia de ASP.NET, Madrid 2002.

Tabla 3. Requisitos de hardware según Ediciones de Visual Studio .Net

Requisito	Enterprise Architect	Enterprise Developer	Professional	Academic
Procesador	PC con procesador de tipo Pentium II, 450 MHz Se recomienda: tipo Pentium III, 600 MHz ¹	Igual	Igual	Igual
RAM	Windows 2000 Professional: 96 MB; Windows 2000 Server: 192 MB; Windows XP Home: 96 MB; Windows XP Professional y Windows Server 2003: 192 MB <i>Se recomienda:</i> 128 MB para 2000 Professional, 256 MB para 2000 Server, 160 MB para XP Home y 256 MB para XP Professional y Server 2003 ¹	Igual	Igual	Igual
Espacio disponible en el disco duro	900 MB en la unidad del sistema y 4,1 GB en la unidad de instalación ²	Igual	Igual	Igual
Sistema operativo	Windows® 2000, Windows XP, Windows Server 2003 o Windows NT 4.0 ^{3, 4, 5}	Igual	Igual	Igual
Unidad de CD-ROM o DVD-ROM	Requerida ⁶	Requerida	Requerida	Requerida
Vídeo	800 x 600, 256 colores Se recomienda: color de alta densidad, 16 bits	Igual	Igual	Igual
Mouse (ratón)	Microsoft Mouse o dispositivo señalador compatible	Igual	Igual	Igual

Fuente: Ayuda MSDN; de Microsoft Visual Studio. Net, año 2003

Requisitos Edición Estándar de Visual Studio .NET

Tabla 4. Requisitos de hardware Edición Estándar de Visual Studio .Net

Requisito	Visual Basic	Visual C++	Visual C#	Visual J#
Procesador	PC con procesador de tipo Pentium II, 450 MHz Se recomienda: tipo Pentium III, 600 MHz ¹	Igual	Igual	Igual
RAM	Windows 2000 Professional: 96 MB; Windows 2000 Server: 192 MB; Windows XP Home: 96 MB; Windows XP Professional y Server 2003: 192 MB <i>Se recomienda:</i> 128 MB para 2000 Professional, 256 MB para 2000 Server, 160 MB para XP Home y 256 MB para XP Professional y Server 2003 ¹	Igual	Igual	Igual
Espacio disponible en el disco duro	750 MB en la unidad del sistema y 2,5 GB en la unidad de instalación ²	Igual	Igual	Igual
Sistema operativo	Windows® 2000, Windows XP, Windows Server 2003 o Windows NT 4.0 ^{3,4,5}	Igual	Igual	Igual
Unidad de CD-ROM o DVD-ROM	Requerida ⁶	Igual	Igual	Igual
Vídeo	800 x 600, 256 colores Se recomienda: color de alta densidad, 16 bits	Igual	Igual	Igual
Mouse (ratón)	Microsoft Mouse o dispositivo señalador compatible	Igual	Igual	Igual

Fuente: Ayuda MSDN; de Microsoft Visual Studio. Net, año 2003

¹El rendimiento no se ha optimizado para la configuración mínima del sistema. Si aumenta la cantidad de RAM más allá de la configuración de sistema recomendada aumentará el rendimiento, especialmente al ejecutar varias aplicaciones, trabajar con proyectos de gran tamaño o hacer desarrollos para empresas.

²Cuando inicia el instalador de Visual Studio .NET, la ubicación de instalación predeterminada es la unidad de sistema, que es la unidad que inicia el sistema. Sin embargo, puede instalar la aplicación en cualquier unidad. Independientemente de dónde se encuentre ubicada la aplicación, el proceso de instalación instala algunos archivos en la unidad de sistema. Por tanto, deberá asegurarse de que el espacio indicado anteriormente se encuentra disponible en la unidad de sistema, aunque la aplicación se encuentre en otra ubicación, y de que también hay espacio disponible, tal y como se ha indicado anteriormente, en la unidad en la que desea instalar la aplicación.

³Windows NT 4.0 sólo admite la instalación de componentes remotos. No se puede instalar Visual Studio con Windows NT 4.0.

⁴Windows XP Home no admite el desarrollo local de aplicaciones Web, solamente las versiones Professional y Server de Windows admiten esta función.

⁵Microsoft Windows 2000 Datacenter Server no es un sistema operativo compatible.

⁶El tipo de medios que se distribuyen con el producto determinan si se necesitará un CD-ROM o un DVD-ROM⁴.

1.1.3. Visual C# .NET.

1.1.3.1. Generalidades de C#.NET, Lenguaje nativo de .NET.

Visual C# (pronunciado "C sharp") está diseñado para crear de manera rápida y fácil aplicaciones .NET, incluyendo servicios Web y aplicaciones Web ASP.NET. Las aplicaciones escritas en Visual C# se generan con los servicios de Common Language Runtime y se benefician de .NET Framework.

C# (leído en inglés "C Sharp" y en español "C Almohadilla") es el nuevo lenguaje de propósito general diseñado por Microsoft para su plataforma .NET. Sus principales creadores son Scott Wiltamuth y Anders Hejlsberg, éste último también conocido por haber sido el diseñador del lenguaje Turbo Pascal y la herramienta RAD Delphi.

⁴ <ms-help://MS.MSDNQTR.2003FEB.3082/vsintro7/html/vxoriVisualStudioNETSystemRequirements.htm>

C# es un lenguaje de programación que toma las mejores características de lenguajes preexistentes como Visual Basic, Java o C++ y las combina en uno solo.

C# es un lenguaje orientado a objetos simple, elegante y con seguridad en el tratamiento de tipos que acaba de implementar Microsoft para generar una gran variedad de aplicaciones. Cualquiera que esté familiarizado con C y lenguajes similares, encontrará pocos problemas para adaptarse a C#. C# está diseñado para que el programador de C++ haga desarrollos rápidos, sin sacrificar la potencia y el control que constituyen el sello de C y C++. Gracias a esta herencia, C# tiene un alto grado de fidelidad con C y C++, y los programadores familiarizados con estos lenguajes pueden ser rápidamente productivos en C#. C# proporciona mecanismos intrínsecos de código de confianza para obtener un nivel alto de seguridad, la recolección de elementos no utilizados y la seguridad de tipos. C# admite herencia única y crea lenguaje intermedio de Microsoft (MSIL) como entrada de compiladores de código nativo.

Aunque es posible escribir código para la plataforma .NET en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella, por lo que programarla usando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes

ya que C# carece de elementos heredados innecesarios en .NET. Por esta razón, se suele decir que C# es el lenguaje nativo de .NET.

La sintaxis y estructuración de C# es muy similar a la C++, ya que la intención de Microsoft con C# es facilitar la migración de códigos escritos en estos lenguajes a C# y facilitar su aprendizaje a los desarrolladores habituados a ellos. Sin embargo, su sencillez y el alto nivel de productividad son equiparables a los de Visual Basic.

C# está completamente integrado con .NET Framework y Common Language Runtime, que conjuntamente proporcionan interoperabilidad del lenguaje, recolección de elementos no utilizados, seguridad ampliada y compatibilidad de versiones mejorada. C# simplifica y moderniza algunos de los aspectos más complejos de C y C++, como los espacios de nombres, las clases, las enumeraciones, la sobrecarga y el control estructurado de excepciones. C# también elimina ciertas características de C y C++ como macros, herencia múltiple y clases base virtuales. Para los programadores de C++ actuales, C# proporciona un lenguaje alternativo de gran potencia y productividad.

Visual C# proporciona prototipos de algunos de los tipos de proyectos más comunes, incluyendo:

- Aplicación para Windows.
- Biblioteca de clases.
- Biblioteca de control de Windows.
- Aplicación Web ASP.NET.
- Servicio Web ASP.NET.
- Biblioteca de control Web.
- Aplicación de consola.
- Servicio de Windows⁵.

1.1.3.2. Características y Beneficios de C#.NET.

Visual C# .NET es un completo conjunto de herramientas para la creación de servicios Web XML y aplicaciones basadas en Microsoft .NET para Microsoft Windows® y el Web. Este eficaz paquete de programación, que utiliza el lenguaje de programación orientado a componentes C#, ofrece a los programadores de nivel inicial o intermedio con conocimientos de C++ o Java un entorno y un lenguaje modernos para la creación de software de próxima generación. Visual C# .NET ofrece funcionalidad superior para optimizar procesos empresariales, por ejemplo:

⁵ ms-help://MS.VSCC.2003/MS.MSDNQTR.2003FEB.3082/vsent7/html/vxconProgrammingLanguages.htm

- Compatibilidad con el diseño, la programación y la implementación de servicios Web XML con rapidez.
- Diseñadores de formularios y controles visuales para crear aplicaciones basadas en Windows muy completas.
- Herramientas y servicios de diseño para crear eficaces soluciones de Microsoft .NET basadas en servidor.

Con Visual C# .NET, los programadores pueden crear soluciones para una amplísima gama de clientes, incluidos Windows, el Web y dispositivos de cliente ligero. Con este elegante y moderno lenguaje de programación, los programadores pueden aprovechar sus conocimientos y habilidades de C++ y Java para disfrutar de una experiencia satisfactoria con la plataforma Microsoft .NET⁶.

1.1.3.3. Sencillez.

C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET. Por ejemplo:

El código escrito en C# es **autocontenido**, lo que significa que no necesita de ficheros adicionales al propio fuente tales como ficheros de cabecera o ficheros IDL.

⁶ UTC; Internet; <http://www.microsoft.com/latam/vcsharp/producto/caracteristicas.asp>

El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile (no como en C++), lo que facilita la portabilidad del código.

No se incluyen elementos poco útiles de lenguajes como C++ tales como macros, herencia múltiple o la necesidad de un operador diferente del punto (.) acceder a miembros de espacios de nombres (::)⁷.

1.1.3.4. Modernidad.

C# incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como Java o C++ hay que simular, como un tipo básico **decimal** que permita realizar operaciones de alta precisión con reales de 128 bits (muy útil en el mundo financiero), la inclusión de una instrucción **foreach** que permita recorrer colecciones con facilidad y es ampliable a tipos definidos por el usuario, la inclusión de un tipo básico **string** para representar cadenas o la distinción de un tipo **bool** específico para representar valores lógicos.

⁷ UTC, Internet; GONZÁLEZ, José Antonio; El lenguaje de programación C#; año 2000

1.1.3.5. Orientación a objetos.

Como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos, aunque eso es más bien una característica del CTS que de C#. Una diferencia de este enfoque orientado a objetos respecto al de otros lenguajes como C++ es que el de C# es más puro en tanto que no admiten ni funciones ni variables globales sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código.

C# soporta todas las características propias del paradigma de programación orientada a objetos: **encapsulación**, **herencia** y **polimorfismo**.

1.1.3.6. Orientación a componentes.

La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas. Es decir, la sintaxis de C# permite definir cómodamente **propiedades** (similares a campos de acceso controlado), **eventos** (asociación controlada de funciones de respuesta a notificaciones) o **atributos** (información sobre un tipo o sus miembros)

1.1.3.7. Instrucciones seguras.

Para evitar errores muy comunes, en C# se han impuesto una serie de restricciones en el uso de las instrucciones de control más comunes. Por ejemplo, la guarda de toda condición ha de ser una expresión condicional y no aritmética, con lo que se evitan errores por confusión del operador de igualdad (==) con el de asignación (=); y todo caso de un **switch** ha de terminar en un **break** o **goto** que indique cuál es la siguiente acción a realizar, lo que evita la ejecución accidental de casos y facilita su reordenación.

1.1.3.8. Eficiencia.

En principio, en C# todo el código incluye numerosas restricciones para asegurar su seguridad y no permite el uso de punteros. Sin embargo, y a diferencia de Java, en C# es posible saltarse dichas restricciones manipulando objetos a través de punteros. Para ello basta marcar regiones de código como inseguras (modificador **unsafe**) y podrán usarse en ellas punteros de forma similar a cómo se hace en C++, lo que puede resultar vital para situaciones donde se necesite una eficiencia y velocidad procesamiento muy grandes⁸.

⁸ UTC, Internet; <http://microsoft.public.dotnet.languages.csharp>.

1.1.3.9. Introducción a las páginas de Formularios Web Forms.

Las páginas de formularios Web Forms pueden usarse para crear páginas Web programables que sirvan como interfaz de usuario de las aplicaciones Web. Este tipo de páginas muestran la información al usuario en cualquier explorador o dispositivo cliente e implementa lógica de aplicación mediante el código de la parte servidor. La salida de las páginas de formularios Web Forms puede contener casi cualquier lenguaje compatible con HTTP, incluidos HTML, XML, WML y ECMAScript (JScript, JavaScript).

Las páginas de formularios Web Forms reúnen las siguientes características:

- Se basan en la tecnología Microsoft ASP.NET, en la que el código que se ejecuta en el servidor genera de forma dinámica salida de páginas Web en un explorador o dispositivo cliente.
- Compatible con cualquier explorador o dispositivo móvil. Las páginas de formularios Web Forms presentan automáticamente el código HTML adecuado al explorador para funciones tales como estilos, diseño, etc. Como alternativa, se pueden diseñar las páginas de formularios Web Forms para ejecutarse en un explorador determinado, como Microsoft Internet Explorer 5 y aprovechar así todas las funciones de un cliente de explorador de nivel superior.

- Admiten cualquier lenguaje compatible con Common Language Runtime de .NET, incluidos Microsoft Visual Basic, **Microsoft Visual C#** y Microsoft JScript.NET.
- Se crean en el entorno Microsoft .NET Framework. Esto proporciona todos los beneficios del marco de trabajo, incluidos un entorno administrado, seguridad de tipos y herencia.
- Respaldadas en Visual Studio por eficaces herramientas de desarrollo rápido de aplicaciones (RAD, *Rapid Application Development*) destinadas al diseño y la programación de los formularios.
- Extensibles mediante controles que proporcionan posibilidades RAD al desarrollo Web, lo que permite crear con rapidez interfaces de usuario enriquecidas.
- Flexibles gracias a la posibilidad de incorporar a ellas controles creados por los usuarios y de otros fabricantes.

1.1.3.9.1. Componentes de los Formularios Web Forms.

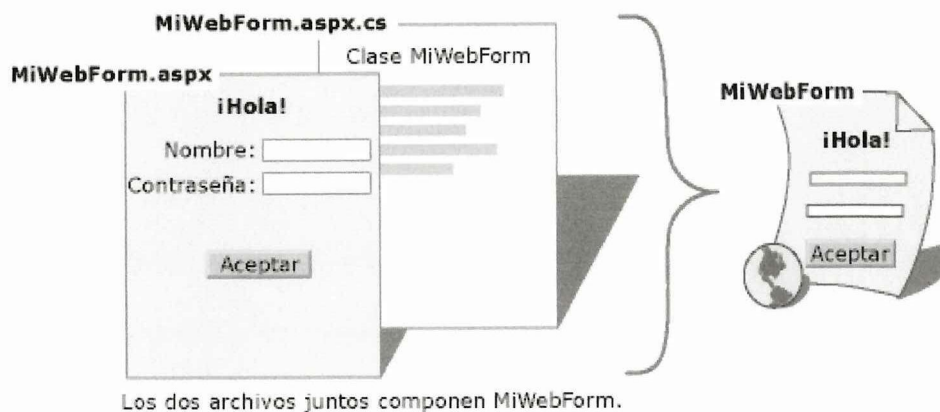
En las páginas de formularios Web Forms, la programación de la interfaz de usuario se divide en dos partes independientes: el componente visual y el lógico. Esta división entre la parte visible del formulario y el código que se oculta detrás y que interactúa con él.

El elemento visual se conoce como la *página* de formularios Web Forms, y se compone de un archivo que contiene código HTML estático, o controles de servidor ASP.NET o ambos de forma simultánea.

La página de formularios Web Forms funciona como un contenedor del texto y los controles estáticos que se desea mostrar. Si se usa el Diseñador de Web Forms de Visual Studio junto con controles de servidor ASP.NET, se pueden diseñar los formularios igual que se haría en cualquier aplicación de Visual Studio.

La lógica de las páginas de formularios Web Forms se compone del código creado para interactuar con el formulario. La lógica de programación reside en un archivo independiente del archivo de la interfaz de usuario. Este archivo se conoce como el archivo de "código subyacente" y adopta la extensión ".aspx.cs". La lógica escrita en el archivo de código subyacente puede estar en Visual Basic o en Visual C#.

Figura 5. Estructura de los archivos de formularios Web Forms



Fuente: Ayuda MSDN; de Microsoft Visual Studio. Net, año 2003

Los archivos de código subyacente de todas las páginas de formularios Web Forms de un proyecto se compilan en el archivo de biblioteca de vínculos dinámicos (.dll) del proyecto. El archivo de página .aspx también se compila, pero de un modo distinto. La primera vez que un usuario examina la página .aspx con el explorador, ASP.NET genera automáticamente un archivo de clase .NET que representa a la página y que la compila en un segundo archivo .dll. La clase generada para la página .aspx hereda de la clase del código subyacente que se compiló en el archivo .dll del proyecto. Cuando un usuario solicita la dirección URL de la página Web, los archivos .dll se ejecutan en el servidor y producen de forma dinámica la salida HTML de la página.

1.1.3.9.2. Ventajas y Beneficios que otorgan las páginas en Formularios Web Forms.

La programación de aplicaciones Web presenta retos que no surgen normalmente en la programación tradicional de aplicaciones basadas en clientes. Entre estos retos se encuentran los siguientes:

1.1.3.9.2.1. Implementar una interfaz de usuario Web enriquecida.

Una interfaz de usuario con un diseño complejo, una gran cantidad de contenido dinámico y llena de objetos interactivos y plenos de funcionalidad puede resultar difícil y tediosa de diseñar e implementar si se utilizan herramientas HTML básicas. Resulta particularmente difícil crear una interfaz de usuario enriquecida para aplicaciones que deban ejecutarse en muchos exploradores y plataformas de dispositivos clientes distintos.

1.1.3.9.2.2. Separación entre Cliente / Servidor.

En las aplicaciones Web, el cliente (explorador) y el servidor son programas distintos que a menudo se ejecutan en equipos distintos e, incluso, en sistemas operativos diferentes. Por lo

tanto, las dos mitades de la aplicación comparten muy poca información; se pueden comunicar, pero normalmente intercambian sólo pequeñas porciones de información simple.

1.1.3.9.2.3. Ejecución Independiente.

Cuando un servidor Web recibe una petición de una página, la busca, la procesa y la envía al explorador y, a continuación, desecha toda la información sobre dicha página. Si el usuario solicita la página de nuevo, el servidor repite la secuencia completa, volviendo a procesar la página desde el principio. En otras palabras, los servidores no tienen memoria de las páginas que han procesado. Por lo tanto, si una aplicación necesita mantener información sobre una página, esto se convierte en un problema que tiene que resolverse en el código de la aplicación.

1.1.3.9.2.4. Compilaciones con el acceso a los datos.

La lectura de los datos de un origen de datos y la escritura en el mismo puede resultar complicada con las aplicaciones Web tradicionales y hacer un gran uso de los recursos.

1.1.3.9.2.5. Compilaciones con la escalabilidad.

En muchos casos las aplicaciones Web diseñadas con los métodos existentes no pueden cumplir los objetivos de escalabilidad debido a la falta de compatibilidad entre sus distintos componentes. Este es a menudo el único origen de los errores en aplicaciones sometidas a un ciclo de crecimiento intenso.

1.1.3.9.3. Derivar de la clase Page.

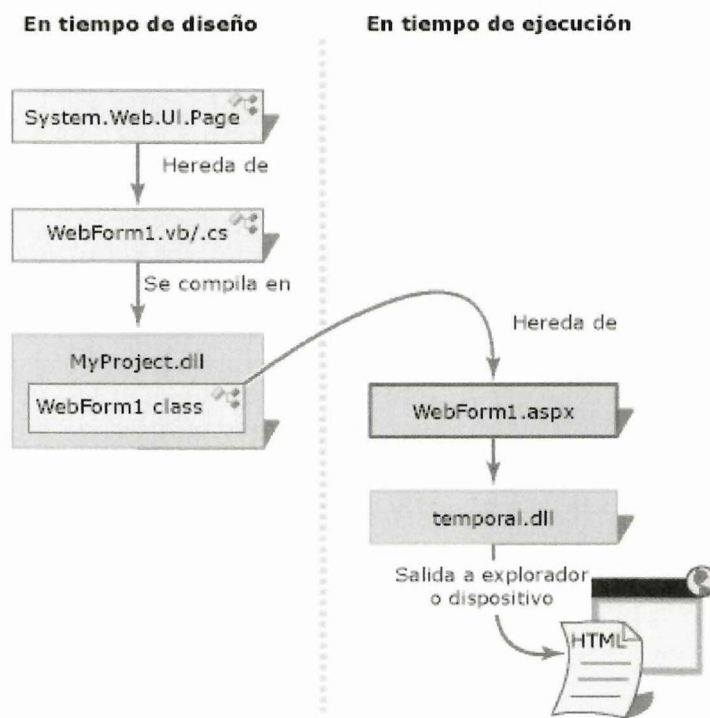
Cuando en Visual Studio crea los archivos de página y de clase de una página de formulario Web Forms, genera código que hereda de la clase **Page** básica. Por ejemplo, si se crea una nueva página de formularios Web Forms y se la denomina `WebPage1`, se derivará una nueva clase denominada `WebPage1` de **System.Web.UI.Page**.

Si se ha elegido Visual C# para el desarrollo, Visual Studio generará una de las siguientes líneas de código:

```
// C#  
  
public class WebForm1 : System.Web.UI.Page
```

A su vez, el archivo de página .aspx se hereda de la clase derivada WebPage1. La relación de la clase **Page** básica, el archivo de clase derivada y el archivo .aspx se muestra en el diagrama siguiente.

Figura 6. Estructura de la página de formularios Web Forms y la clase Page básica



Fuente: Ayuda MSDN; de Microsoft Visual Studio. Net, año 2003

Debido a que el archivo .aspx se compila de forma dinámica cuando un usuario examina la página, su relación con el archivo de clase se establece mediante directivas de secuencias de comandos en la parte superior de la página. En Visual Studio, la relación entre los archivos

.aspx y los archivos de clase se crea y mantiene automáticamente, incluso si se cambia el nombre de la página de formularios Web Forms. De forma específica, se usa el atributo **Inherits** de la directiva **@ Page** para determinar el archivo de clase del que se derivará el archivo .aspx. Una directiva típica tiene este aspecto:

```
<%@ Page Language="cs" AutoEventWireup="false"  
Codebehind="WebForm1.aspx.cs"  
Inherits="WebApplication1.WebForm1"%>
```

1.1.3.9.4. Elementos de las aplicaciones WEB ASP.NET.

1.1.3.9.4.1. Introducción a ASP.NET.

ASP.NET es más que una nueva versión de las páginas Active Server (ASP); proporciona un modelo de desarrollo Web unificado que incluye los servicios necesarios para que los programadores creen aplicaciones Web para la empresa. Si bien ASP.NET es en gran medida compatible con la sintaxis de ASP, proporciona también un modelo de programación y una nueva estructura para crear aplicaciones más escalables y estables que ayuden a proporcionar mayor protección. Las aplicaciones ASP se pueden ampliar agregándoles funcionalidad de ASP.NET.

ASP.NET es un entorno compilado basado en .NET. Se pueden crear aplicaciones en cualquier lenguaje compatible con .NET, como Visual Basic .NET, C# y JScript .NET. Además, .NET Framework está disponible en su totalidad para cualquier aplicación ASP.NET. Los programadores pueden aprovechar fácilmente las ventajas de estas tecnologías, que incluyen el entorno Common Language Runtime administrado, seguridad de tipos, herencia, etc.

La creación de aplicaciones Web ASP.NET implica trabajar con muchos de los elementos que se utilizan en cualquier aplicación de escritorio o cliente-servidor. Éstos incluyen:

1.1.3.9.4.2. Funciones de administración de proyectos.

Al crear una aplicación Web ASP.NET, necesitará hacer un seguimiento de los archivos necesarios, cuáles se debe compilar y cuáles necesitan implementarse.

1.1.3.9.4.3. Interfaz de usuario.

Una aplicación suele presentar información a los usuarios; en una aplicación Web ASP.NET, la interfaz de usuario se presenta en páginas de formularios Windows Forms, que envían los

resultados a un explorador. Opcionalmente, puede crear resultado adaptado para dispositivos móviles u otros aparatos Web.

1.1.3.9.4.4. Componentes.

Muchas aplicaciones incluyen elementos reutilizables que contienen código para ejecutar tareas específicas. En las aplicaciones Web, puede crear estos componentes como servicios Web XML, lo que les permite aceptar llamadas de todo el Web desde una aplicación Web, otro servicio Web XML o un formulario Windows Forms.

1.1.3.9.4.5. Datos.

La mayoría de las aplicaciones necesitan algún mecanismo de acceso a datos. En las aplicaciones Web ASP.NET, puede utilizar ADO.NET, los servicios de datos que forman parte de .NET Framework.

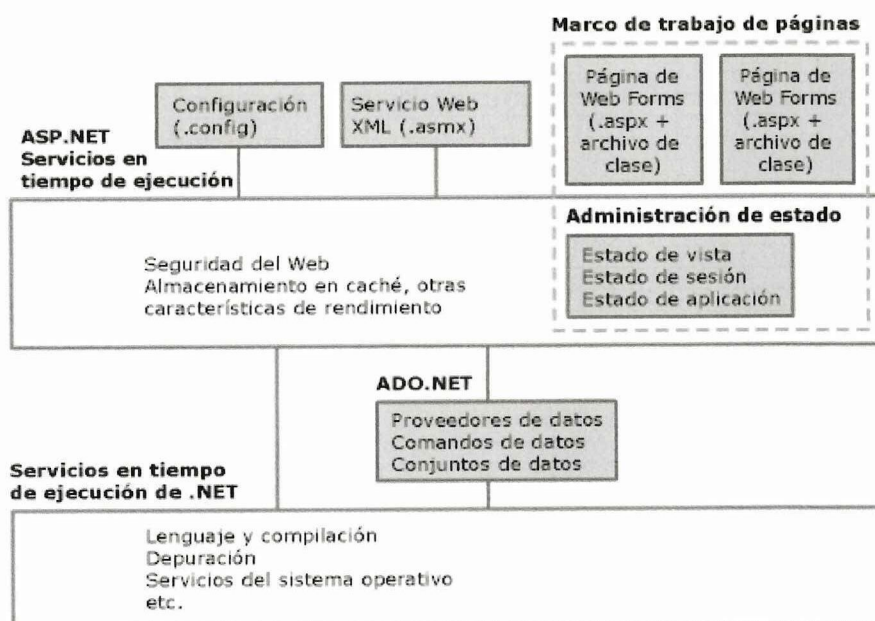
1.1.3.9.4.6. Seguridad, rendimiento y otras características de infraestructura.

Como en cualquier aplicación, deberá implementar seguridad para evitar el uso no autorizado, probar y depurar la aplicación, ajustar

su rendimiento y ejecutar otras tareas no relacionadas con la función principal de la aplicación.

La Figura 7, proporciona información general sobre el modo en que encajan entre sí las diferentes piezas de las aplicaciones Web ASP.NET y sobre cómo encajan en el contexto más amplio de .NET Framework⁹.

Figura 7. Contexto de .NET Framework



Fuente: Ayuda MSDN; de Microsoft Visual Studio. Net, año 2003

⁹ UTC, Internet; <http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/cpguide/html/cpconintroductiontoasp.asp>

1.1.4. ADO NET.

Con la necesidad de la conexión a bases de datos surge una implementación en .NET conocida como ADO.NET que pretende la comunicación con bases de datos a un alto nivel de abstracción. Va a permitir la manipulación de bases de datos, agregando, insertando, borrando o modificando registros¹⁰.

Con la implementación de ADO.NET se consigue la manipulación de bases de datos como ORACLE, PostgreSQL, MySQL, SQL, SQLite entre otras, al igual como OLEDB entre otros proveedores. Esto permite la conectividad a las bases de datos de una forma sencilla utilizando una clase para la conexión pertinente en la que se le pasa el proveedor, usuario, contraseña y la ubicación de la base de datos con la que queremos conectar. Conforme vayan apareciendo nuevos proveedores se irán implementando nuevas clases en libgda y aumentará el número de tipos diferentes de bases de datos a las que podremos conectarnos.

De igual forma que la lectura y manipulación de una base de datos tendremos otras características como la creación y ejecución de procedimientos almacenados, como también la manipulación de bases de datos desconectadas conocidas como DataSets.

¹⁰ UTC, Internet; <http://www.monohispano.org/tutoriales/hispalinux2002/x136.html>

1.1.4.1. ADO NET Características.

ADO.NET es un conjunto de clases que exponen servicios de acceso a datos al programador de .NET. ADO.NET proporciona un conjunto variado de componentes para crear aplicaciones distribuidas de uso compartido de datos. Forma parte integral de .NET Framework, y proporciona acceso a datos relacionales, datos XML y datos de aplicaciones. ADO.NET es compatible con diversas necesidades de programación, incluida la creación de clientes de bases de datos clientes y objetos empresariales de nivel medio utilizados por aplicaciones, herramientas, lenguajes o exploradores de Internet¹¹.

1.1.4.2. Interacciones con base de datos.

Las interacciones con la base de datos se realizan mediante comandos de datos. Para efectuar operaciones en una base de datos, se ejecutan instrucciones SQL o procedimientos almacenados (que incluyen instrucciones SQL). Las instrucciones SQL o los procedimientos almacenados se usan para leer y escribir en filas y para ejecutar funciones agregadas, como la adición o la obtención de un promedio. Asimismo, se utilizan para crear o modificar tablas o columnas, realizar transacciones, etc.

¹¹ ms-help://MS.MSDNQTR.2003FEB.3082/cpguide/html/cpconaccessingdatawithadonet.htm

1.1.4.3. ADO.NET no depende de conexiones continuamente activas.

En las aplicaciones tradicionales cliente/servidor, los componentes establecen una conexión con una base de datos y la mantienen abierta mientras la aplicación está en funcionamiento. Por diferentes razones, este enfoque no resulta práctico en otras aplicaciones:

- Una conexión abierta de base de datos consume recursos valiosos del sistema. En la mayoría de los casos, las bases de datos sólo pueden mantener un pequeño número de conexiones concurrentes. La sobrecarga que supone mantener estas conexiones reduce el rendimiento general de la aplicación.
- Las aplicaciones que necesitan una conexión de base de datos abierta resultan muy difíciles de ampliar. Una aplicación difícil de ampliar puede funcionar aceptablemente con cuatro usuarios, pero no es probable que lo haga con cientos. Las aplicaciones Web ASP.NET, deben ser fácilmente escalables, porque el tráfico de un sitio Web puede crecer en órdenes de magnitud en un periodo de tiempo muy corto.
- En las aplicaciones Web ASP.NET, los componentes están desconectados entre sí inherentemente. El explorador solicita una página al servidor; al terminar el servidor de procesar y enviar la

página, no tiene ninguna conexión con el explorador hasta la próxima petición. Bajo estas circunstancias, el mantenimiento de conexiones abiertas con una base de datos no es viable, porque no hay modo de saber si el consumidor de datos (el cliente) continúa necesitando acceso a los datos.

Por todas estas razones, el acceso a datos con ADO.NET se diseñó en torno a una arquitectura que utiliza conexiones con moderación. Las aplicaciones se conectan a la base de datos sólo durante el tiempo necesario para extraer o actualizar los datos. La base de datos ya no contiene conexiones que la mayor parte del tiempo están inactivas, así que puede dar servicio a muchos más usuarios.

1.1.4.4. Los conjuntos de datos son independientes de los orígenes de datos.

Aunque un conjunto de datos actúa como una memoria caché para datos extraídos de una base de datos, el conjunto de datos no tienen ninguna relación real con la base de datos. El conjunto de datos es un contenedor; se llena con comandos SQL o procedimientos almacenados que se ejecutan desde un adaptador de datos.

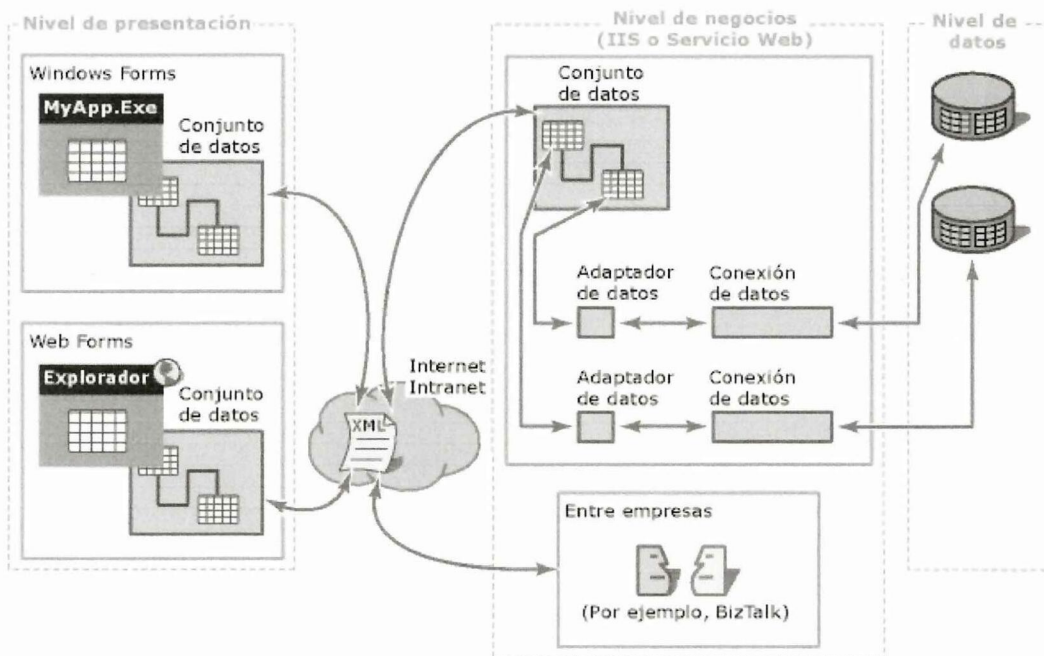
Dado que un conjunto de datos no está sujeto directamente a un origen de datos, resulta un buen punto de integración para datos procedentes de múltiples orígenes. Por ejemplo, algunos datos de un conjunto de datos

podrían proceder de su base de datos, mientras que otros podrían venir de una base de datos diferente o de un origen distinto a una base de datos como, por ejemplo, una hoja de cálculo. Parte de los datos de un conjunto de datos podrían llegar en una secuencia enviada por otro componente.

1.1.4.5. Componentes de ADO.NET.

En la Figura 7, se puede apreciar los principales componentes de una aplicación ADO.NET.

Figura 8. Componentes de ADO.NET



Fuente: Ayuda MSDN; de Microsoft Visual Studio. Net, año 2003

1.2. SISTEMAS OPERATIVOS PARA SERVIDORES

1.2.1. Sistema Operativo Windows XP Profesional.

1.2.1.1. Introducción.

Windows XP proporciona las herramientas necesarias para trabajar en el entorno actual de comunicaciones de alta velocidad. Es la versión actualizada y mejorada de Windows 2000 Profesional y Windows Millennium Edition, está diseñado para servir como concentrador central de información para servicios y actúa como el dispositivo más inteligente de entre un conjunto en expansión de PC de bolsillo, teléfonos móviles, PC de tableta, cámaras digitales y otros dispositivos. Incluye herramientas de sistema que le ayudan a configurar y ajustar recursos del mismo, grupos de usuarios y tareas programadas del sistema. Puede crear documentos e imágenes, calcular cifras, programar su tiempo y celebrar reuniones en línea con sus colaboradores. Windows XP Profesional está diseñado para proporcionar una base fiable que proporcione lo más reciente en seguridad y confidencialidad, al tiempo que ofrece excepcional rendimiento y facilidad de uso. Las comunicaciones y las herramientas de entretenimiento le ayudarán a relajarse.

1.2.1.2. Características de Windows XP Profesional.

Entre sus nuevas características están:

- Es desarrollado en base al código comprobado de Windows NT® y Windows 2000, que presenta una arquitectura de cómputo de 32 bits, así como un modelo de memoria totalmente protegido.
- Se puede compartir la conexión de Internet con todos los equipos de la red, con la seguridad que el equipo está protegido por un Servidor de seguridad de conexión a Internet.
- Permite comunicarse con otras personas en otras redes sin tener que preocuparse por comprometer la privacidad o la seguridad de archivos personales.
- El rendimiento es alto en todo momento, está diseñado para permitir que varias aplicaciones se ejecuten simultáneamente, al tiempo que asegura un excelente tiempo de respuesta y estabilidad del sistema.
- Es más confiable y estable, por que siempre podrá contar con que la PC funcione de forma eficaz y a pleno rendimiento.
- La compatibilidad de los programas es mejor que nunca.
- Facilita la configuración y administración de las cuentas de equipo a cualquier usuario que utilice el PC.

- Facilita la creación de redes domésticas, pudiendo utilizar el asistente para configuración de red para crear rápidamente su propia red.
- Escritorio Remoto, es una de las nuevas características tecnológicas que brinda Windows XP, que permite tener acceso a una sesión de Windows desde otro equipo estando en otro sitio y siempre que se encuentre en red.
- Protección de archivos y Restaurar Sistema, evitarán que pueda eliminar accidentalmente archivos importantes o del sistema y devolverán al equipo a su estado anterior si se produce algún error grave.
- Es posible trabajar con archivos y carpetas de forma más eficaz, visualizando con una interfaz de página Web más intuitiva e informativa.
- Brinda opciones para asociar ciertos tipos de archivos a programas específicos y trabajar sin conexión con archivos y carpetas es más sencillo y dinámico.
- Permite compresión y encriptación de archivos y carpetas rápida y sencillamente. Así ocuparán menos espacio de almacenamiento en el equipo.

- Contiene un administrador de credenciales que es un depósito seguro para información de claves de acceso, permite ingresar el nombre de usuario y las claves de varios recursos y aplicaciones en la red (como correo electrónico) una vez y luego hacer que el sistema proporcione automáticamente esa información cuando vuelva a utilizar esos recursos

1.2.2. Sistema Operativo Microsoft Windows 2003 Server.

1.2.2.1. Introducción.

Windows Server 2003 es un sistema operativo de propósitos múltiples capaz de manejar una gran gama de funciones de servidor, en base a sus necesidades, tanto de manera centralizada como distribuida.

Windows Server 2003 contiene tecnologías básicas construidas en base a las fortalezas de Windows 2000 Server para ofrecer un sistema operativo rentable y superior. Windows Server 2003 es una plataforma de servidor ideal para organizaciones de cualquier tamaño¹².

Algunas de estas funciones del servidor son:

- Servidor de archivos e impresión.
- Servidor Web y aplicaciones Web.
- Servidor de correo.

¹² UTC; Internet; <http://www.monografias.com/trabajos14/microsoftwindows/microsoftwindows.shtml>

- Terminal Server.
- Servidor de acceso remoto/red privada virtual (VPN).
- Servicio de directorio, Sistema de dominio (DNS), y servidor DHCP.
- Servidor de transmisión de multimedia en tiempo real (Streaming).
- Servidor de infraestructura para aplicaciones de negocios en línea (tales como planificación de recursos de una empresa y software de administración de relaciones con el cliente).

1.2.2.2. Características de Windows 2003 Server.

Entre las características de Windows 2003 Server están:

- Es el Sistema Operativo de servidor más rápido y seguro que ha existido. Ofrece fiabilidad, disponibilidad y escalabilidad al proporcionar una infraestructura integrada y de red que asegura la información.
- Ofrece herramientas que permiten implementar, administrar y usar una infraestructura de red para obtener una productividad máxima.
- Ayuda a crear una infraestructura de soluciones de negocios para mejorar la conectividad con empleados, socios, sistemas y clientes.

- Proporciona un servidor Web integrado y un servidor de transmisión de multimedia en tiempo real para crear de forma fácil, rápida, segura una Intranet dinámica y sitios de Internet.
- Proporciona un servidor de aplicaciones integrado para desarrollar, implementar y administrar servicios Web en XML.
- En el campo económico, al estar combinado con productos Microsoft en el hardware y software, ayuda a obtener el rendimiento más alto en sus inversiones.
- Como Servidor Web es de un 100% a un 165% más rápido que Windows 2000 Server.
- Combinado con Visual Studio .NET 2003, se convierte en la plataforma más productiva para implementar, ejecutar y gestionar aplicaciones conectadas mediante la nueva generación de servicios Web basados en ASP y XML.
- Integra Internet Information Services (IIS), totalmente rediseñado con el objetivo de mejorar la seguridad, fiabilidad y rendimiento, se instala completamente bloqueado por defecto. Basado en una nueva arquitectura, las aplicaciones Web en ejecución están aisladas una de otra, permitiéndose la monitorización y administración proactiva de aplicaciones así como cambios de configuración en línea, reduciendo el tiempo que precisan los

administradores para reiniciar servicios con el fin de mantener las aplicaciones operativas.

1.3. BASES DE DATOS

1.3.1. Concepto de Bases de datos.

Una base de datos es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su uso posterior. Por ejemplo una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

Desde el punto de vista informático, una base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulan ese conjunto de datos.

Desde el punto de vista más formal, una base de datos es un conjunto de datos estructurados, fiables y homogéneos, organizados independientemente en una máquina, accesibles en tiempo real, compartibles por usuarios concurrentes que tienen necesidades de información diferente y no predecibles en el tiempo.

Las bases de datos cumplen las siguientes propiedades:

- Están estructurados independientemente de las aplicaciones y del soporte de almacenamiento que los contiene.
- Presentan la menor redundancia posible.
- Son compartidos por varios usuarios y/o aplicaciones.

1.3.2. RDBMS.

1.3.2.1. Que es un RDBMS.

Un **RDBMS** es un **Sistema Administrador de Bases de Datos** Relacionales. RDBMS viene del acrónimo en inglés *Relational Data Base Manager System*. Los RDBMS proporcionan el ambiente adecuado para gestionar una base de datos.

CHARTE O. Francisco. Describe: “Las aplicaciones actuales nunca manipulan directamente la información contenida en una base de datos, sino que usan un intermediario, conocido como RDBMS (*Relational Database Management System*), al que envían instrucciones sobre lo que debe hacer. El RDBMS traduce esas instrucciones en acciones concretas: Creación de una tabla, extracción de todas las filas que cumplan un cierto criterio, eliminación de un dato y los relacionados,

etc. También se conoce a un RDBMS como *servidores de datos*¹³ (2002, pp. 540).

1.3.3. Concepto de SQL Server.

SQL (Structured Query Language) Server es un lenguaje de consulta estructurada. Un lenguaje para definir la estructura y procesamiento de una base de datos relacional. Se emplea como un lenguaje de consulta único o puede incorporarse en programas de aplicación. El American National Standards Institute acepta un SQL como una norma en Estados Unidos y fue desarrollado por IBM¹⁴.

1.3.3.1. Introducción a SQL Server y a las Bases de Datos

Relacionales.

Microsoft SQL Server constituye un lanzamiento determinante para los productos de bases de datos de Microsoft, SQL Server es el RDBMS de elección para una amplia gama de clientes corporativos y Proveedores Independientes de Software (ISVs) que construyen aplicaciones de negocios. Las necesidades y requerimientos de los clientes han llevado a la creación de innovaciones de producto significativas para facilitar la utilización, escalabilidad, confiabilidad y almacenamiento de datos.

¹³ UTC, Biblioteca; CHARTE, O. Francisco; Programación con Visual C#. NET; Madrid, 2002

¹⁴ Curso; CASTRO, F. Lorenzo; UCLM-ESI; Modelos de Datos; México, 1999/2000

Una **Base De Datos Relacional** es una base de datos en donde todos los datos visibles al usuario están organizados estrictamente como tablas de valores, y en donde todas las operaciones de la base de datos operan sobre estas tablas.

Estas bases de datos son percibidas por los usuarios como una colección de relaciones normalizadas de diversos grados que varían con el tiempo.

El modelo relacional representa un sistema de bases de datos en un nivel de abstracción un tanto alejado de los detalles de la máquina subyacente, de la misma manera como, por ejemplo, un lenguaje del tipo de PL/1 representa un sistema de programación con un nivel de abstracción un tanto alejado de los detalles de la máquina subyacente. De hecho, el modelo relacional puede considerarse como un lenguaje de programación mas bien abstracto, orientado de manera específica hacia las aplicaciones de bases de datos¹⁵.

1.3.3.2. Características de SQL Server.

SQL Server es la oferta completa de base de datos y análisis. Tanto por la capacidad para consultar la base de datos mediante un explorador como por la compatibilidad con el Lenguaje de marcado extensible (XML, *Extensible Markup Language*), SQL Server es la base de datos

¹⁵ Internet; <http://www.monografias.com/trabajos5/basede/basede.shtml>

totalmente habilitada para Web. Además, ostenta marcas de referencia en cuanto a escalabilidad y confiabilidad, que son críticas para el éxito de una base de datos empresarial. Si lo que se mide es la velocidad en el desarrollo de aplicaciones y la velocidad del procesamiento de transacciones, SQL Server es la base de datos más rápida, convirtiéndole en la opción principal para la empresa que busca agilidad en sus operaciones.

Las características por operación se detallan:

- **Habilitado por completo al Web.**
 - **Compatibilidad con XML.-** Simplifica la integración de sistemas de servicios de fondo y la transferencia de datos.
 - **Análisis habilitado para Web.-** Analiza datos desde sitios remotos accesibles a través de Web.
 - **Acceso Web a los datos.-** Conecta las bases de datos de SQL Server y cubos OLAP de manera flexible, mediante el Web sin necesidad de ninguna programación adicional.
 - **Alojamiento de aplicaciones.-** SQL Server permite aprovechar totalmente las inversiones en hardware de forma que múltiples aplicaciones se pueden ejecutar en un solo servidor o externamente.

- **Seguridad.-** SQL Server garantiza que las aplicaciones sean seguras en cualquier entorno de red, con la seguridad basada en funciones y el cifrado de archivos y de la red.

- **Escalable y confiable.**
 - **Escalabilidad.-** Puede escalar aplicaciones hasta 32 CPU y 64 gigabytes (GB) de RAM. SQL Server ha demostrado un rendimiento insuperable.
 - **Vistas con particiones distribuidas.-** Permite repartir carga entre varios servidores para obtener más escalabilidad.
 - **Alta disponibilidad.-** Aumenta la disponibilidad de aplicaciones empresariales con el trasvase de registros, las copias de seguridad en línea y los clústeres de conmutación por error.
 - **Duplicación.-** Implementa duplicaciones de mezcla, transaccionales y de instantáneas con sistemas heterogéneos.

- **Menor plazo de comercialización.**
 - **Administración de bases de datos simplificada.-** Las características de ajuste y mantenimiento automáticos permiten a los administradores centrar su atención en otras tareas críticas.

- **Productividad del programador mejorada.**- Las funciones definidas por el usuario, la integridad referencial en cascada y el depurador integrado de Transact SQL permiten la reutilización del código para simplificar el proceso de desarrollo.
- **Servicios de transformación de datos.**- Automatiza las rutinas de extracción, transformación y carga de datos de orígenes heterogéneos.
- **Alojamiento Web y de aplicaciones.**- Puede alojar soluciones de comercio electrónico en otros proveedores, gracias a la compatibilidad de SQL Server con varias instancias.
- **English Query (Query Analyser).**- Permite que usuarios planteen preguntas en inglés en lugar de utilizar expresiones multidimensionales¹⁶.

1.3.3.3. SQL Server y el Modelo Cliente / Servidor.

Arquitectura de base de datos cliente servidor.- La estructura de un sistema de cómputo en red, en el cual una computadora, ejecuta servicios a nombre de las otras computadoras. Para un sistema de base de datos, la computadora servidora se denomina servidor de base de datos y procesa el DBMS, así como las computadoras clientes procesan

¹⁶ Internet; <http://www.microsoft.com/latam/sql/evaluation/features/default.asp>

los programas de aplicaciones. Todas las aplicaciones de la base de datos las ejecuta el servidor de la base de datos.

Cliente/Servidor es un método de distribución de información o de archivos en el cual la agrupación central, servidor, almacena los archivos y los hace disponibles para solicitudes de aplicaciones cliente.

SQL Server es la parte del servidor; hay varios clientes de donde elegir para conectarse a SQL Server, incluyendo las utilerías que vienen con éste, como el Analizador de consultas de SQL Server (SQL Server Query Analyzer).

SQL Server proporciona las siguientes ventajas tanto para clientes como para servidores.

1.3.3.3.1. Ventajas de los clientes.

Los clientes proporcionan la interfaz de usuario y puede contener informes, consultas y formularios vacíos.

Las ventajas que ofrecen los clientes son:

- Facilidad de uso.
- Maneja múltiples plataformas de hardware.
- Maneja múltiples aplicaciones de software.
- Familiar al usuario.

1.3.3.3.2. Ventajas del Servidor.

El servidor de la aplicación provee de servicios o ventajas como:

- Seguridad.
- Confiabilidad.
- Concurrencia.
- Tolerancia a fallas.
- Bloqueo sofisticado.
- Hardware de alto rendimiento.
- Control centralizado.

En el modelo cliente/servidor, cuando se ejecuta una consulta, el servidor examina la base de datos y envía al cliente sólo las filas que corresponden. Esto no sólo ahorra tráfico en el ancho de banda de la red, sino que puede ser más rápido que hacer que las estaciones de trabajo realicen la consulta, siempre que el servidor sea una máquina lo suficientemente poderosa.

CAPÍTULO

II

**TRATAMIENTO DE LA INFORMACIÓN DE LA
ACTIVIDAD COMERCIAL DEL ALMACEN
“MULTICOMERCIO CENTRO” DE LA CIUDAD DE
LATACUNGA**

**TRATAMIENTO DE LA INFORMACIÓN DE LA ACTIVIDAD COMERCIAL
DEL ALMACEN “MULTICOMERCIO CENTRO” DE LA CIUDAD DE
LATACUNGA**

2.1. ALMACEN “MULTICOMERCIO CENTRO”.

2.1.1 Datos generales del Almacén “Multicomercio Centro”.

“Multicomercio Centro” se constituye en 1996, surge de la pujanza comercial en su etapa inicial que en ese entonces la ciudad de Latacunga brindaba fruto del adelanto ciudadano y la necesidad de adquisición de bienes materiales para el hogar a precios módicos y con facilidades de pago acorde a la agitada jornada diaria de sus coterráneos.

Inicialmente “Multicomercio Centro” abre sus puertas en un local de la familia Pérez, ubicado en las calles Juan Abel Echeverría y Quito, ofreciendo muebles para el hogar e incursionando con la línea de electrodomésticos en general, siendo estos últimos de menor venta por la competencia ante las grandes casas comerciales, que empezaban a pulular en la ciudad, pero esto no fue obstáculo para que se desarrolle, se engrandezca y necesite otro lugar de mayor espacio, es por ello que aprovecha la oportunidad de tomar en arriendo un local perteneciente a la familia Cajiao, ubicado en las calles Félix Valencia y Belisario Quevedo y por último crear una sucursal en un local de la familia Razo, ubicado en las calles Félix Valencia, entre Quito y Belisario Quevedo.

Conforme pasa el tiempo “Multicomercio Centro” ha ido evolucionando vertiginosamente a grandes pasos y en la actualidad cuenta con una sucursal para la mejor atención de sus clientes, ofreciendo gran variedad de productos y artículos de la línea para el hogar.

2.1.2 Misión.

Comercializar productos y artículos de la línea para el hogar, encaminado en un marco competitivo de excelente atención al cliente, precios económicos, facilidades de pago accesibles a todo el público, valores éticos y morales, potenciando el crecimiento económico social de sus clientes y de la ciudad en general.

2.1.3 Visión.

Consolidarse como una empresa comercial con un margen de ventas tal, que la competencia no pueda igualar, utilizando la estrategia de ofrecer productos y artículos de buena calidad y un excelente servicio al cliente. Pudiendo establecerse como número uno en el mercado Latacungueño en cuanto a la comercialización de artículos para el hogar.

2.1.4 Objetivos:

- Expende artefactos y enseres de la mejor calidad de la línea para el hogar.

- Brindar un gran stock de productos y artículos para el hogar de acuerdo a la demanda actual de la clientela.
- Potenciar el desarrollo comercial de la ciudad de Latacunga, incrementando locales en el ramo.
- Brindar facilidades para la adquisición de artefactos y productos para el hogar a los clientes.

2.1.5 Política.

Brindar a los clientes una buena atención y excelente trato por parte de los empleados, administrador y todos quienes laboran en “Multicomercio Centro”, además proporcionar la información detallada y veraz sobre los precios, características y detalles que presentan cada uno de los productos o artículos en expendio, así como también las facilidades en cuanto al pago en los períodos de tiempo convenidos entre los clientes y el local comercial.

Los empleados de la empresa comercial también deben verificar que los productos o artículos, sean entregados por parte de los proveedores mayoristas, en óptimas condiciones para ser comercializados, y al ser adquiridos por los consumidores finales o clientes, deben ser entregados en las mejores condiciones para satisfacción de las necesidades del cliente y del local comercial.

2.1.6 Estrategias.

- a) Incursionar en el campo tecnológico, con el fin de obtener ventajas competitivas, publicitarias y de mercado frente a locales comerciales similares.
- b) Explotar la mano de obra calificada y laboral del entorno de nuestra provincia, a fin de obtener de ellos una producción de buena calidad y a bajos costos.
- c) Exponer los productos y artículos en expendio a precios accesibles para la mayoría de las personas, de forma que sean de cómoda adquisición.
- d) Brindar plazos prudenciales y acordes a la economía actual, en la cancelación de las cuotas establecidas entre el cliente y el local comercial.
- e) Realizar estudios sobre Marketing y publicidad, a fin de a conocer la razón social y de existencia de “Multicomercio Centro”.

2.1.7 Recursos.

2.1.7.1 Humanos.

- Clientes.
- Administrador propietario del local comercial.
- Vendedores.

2.1.7.2 Infraestructura.

El local comercial “Multicomercio Centro” tiene sus instalaciones en la ciudad de Latacunga.

- Matriz: Ubicada en las calles Félix Valencia y Belisario Quevedo
- Sucursal: Ubicada en las calles Félix Valencia y Quito

2.1.7.3 Económicos.

Los almacenes “Multicomercio Centro”, tienen su financiamiento directo de su propietario Dr. Gustavo Rodrigo Guacho S. quien desde su inicio contó con capital propio.

2.1.7.4 Tecnológicos.

Los recursos en el ámbito tecnológico con los que actualmente cuentan los almacenes “Multicomercio Centro”, Se detalla enseguida:

➤ **Un computador de escritorio de las siguientes características técnicas:**

- Procesador Intel Pentium IV 3.2 Ghz Socket 775.
- Tarjeta de video, sonido, red, fax modem 56.6 Kbps.
- Tarjeta madre Pchips P13g+, Chipset Intel, 4 Usbs Posteriores.
- Memoria Ram Markvision 256 Mb Pc400.
- Disco Duro Samsung 80 Gb 7200 Rpm.
- CD Rewriter Samsung.
- Floppy Disc.
- Fuente 500w dos tomas USB delanteras.
- Teclado full multimedia, mouse óptico, parlantes stereo.

- Monitor LG 17" Normal Crt Negro controles digitales.

➤ **Un computador portátil de las siguientes características técnicas:**

- Computador IBM con procesador Intel Pentium IV CPU 2.6 Ghz.
- Tarjeta de video, sonido, red, fax modem 56.6 Kbps.
- Memoria Ram 512 Mb.
- Disco Duro 80 Gb.
- CD Rewriter.
- Floppy Disc.
- 2 tomas USB posteriores y 2 tomas laterales.
- Teclado full multimedia, mouse óptico, parlantes incorporados.
- Monitor 15".

2.1.7.5 Recursos tecnológicos implementados.

En el marco de desarrollo del trabajo de tesis, se requirió implementar algunos recursos tecnológicos, mismos que se describen seguidamente:

Se requirió implementar una pequeña red interna (intranet) para acceder a la aplicación desde una máquina que hace las veces de servidor hacia otra que desempeña de cliente, para ello se usaron los siguientes recursos:

- Un swith de 8 puertos.
- Cable UTP cat5 para conectividad entre computadores.

- Conectores RJ-45.
- Ponchadora para RJ-45.

2.2. EL MANEJO DE LA INFORMACIÓN EN LOS ALMACENES

“MULTICOMERCIO CENTRO”.

El manejo de la información en los almacenes “Multicomercio Centro”, se lo hace de forma manual y documentada, siendo ésta una forma ineficaz de llevar a cabo la manipulación de los datos y convirtiéndose en un método violable de la información.

Para la venta a crédito de cualquier producto o artículo que se expende en el almacén, el cliente necesita obligatoriamente una persona que haga las veces o le respalde legalmente como garante, con su respectiva cédula de ciudadanía, posteriormente se llena un formulario en el cual se detalla toda la información tanto del cliente como del garante, siendo el encabezado de estos datos: el número de cédula, nombres y apellidos completos, dirección exacta del domicilio, teléfono, ocupación, dirección exacta del lugar de trabajo, referencias adicionales y observaciones.

Adicional a esta información, se procede a detallar los datos del producto o artículo objeto de la negociación y tratamiento del crédito, dichos datos son: fecha de contrato, número de meses de plazo, fecha de vencimiento, nombre o

descripción del artículo, marca, modelo, serie, color, valor total, pago previo o entrada, saldo y observaciones.

Cabe señalar que “Multicomercio Centro”, para el caso de las ventas a crédito, trabaja con Letras de Cambio, por lo tanto el cliente como su garante deben firmar en los mencionados documentos legales por el valor establecido y el respectivo número de pagos mensuales acordados, por lo que luego de cada pago mensual del cliente, se anota cada vez el valor correspondiente de cada Letra de Cambio, y se procede a la entrega de la misma, dando a conocer al cliente su respectivo saldo y detalle de deuda, para el manejo transparente y eficaz de la información.

En caso de incumplimiento en los pagos, se procede a la notificación verbal por parte del Administrador a los clientes morosos, esperando la pronta respuesta o pronunciamiento, en caso de no existir dicha respuesta, se procede a entregar la documentación legal al Abogado para que éste se encargue de los trámites legales pertinentes para proceder al cobro total del saldo adeudado o el retiro del artículo o producto en litigio.

CAPÍTULO

III

ANÁLISIS DE RESULTADOS

ANÁLISIS DE RESULTADOS

3.1. CRITERIO DEL PROPIETARIO – ADMINISTRADOR.

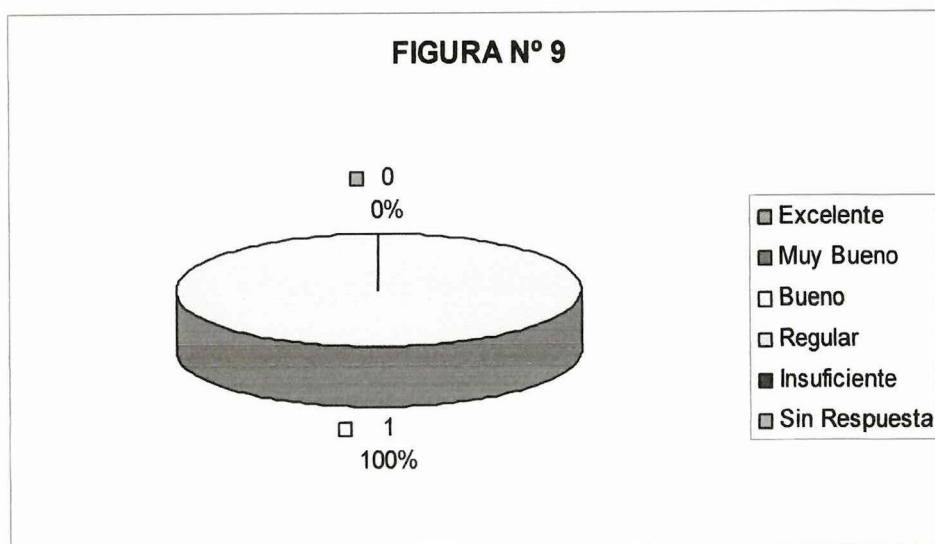
Para realizar la presente investigación se ha desarrollado entrevistas y encuestas dirigidas a cada uno de los recursos humanos que intervienen en el funcionamiento de los almacenes “Multicomercio Centro”, de la ciudad de Latacunga, este el caso del Propietario - Administrador quién está en total acuerdo en dar la apertura para la realización del presente tema de tesis y en vista de la necesidad que existe para dar una mejor atención a los clientes, brinda todo el apoyo necesario y la accesibilidad a toda la información y la forma como se la manipulan los datos en sus locales comerciales, por lo tanto se detalla los resultados a continuación:

PRIMERA PREGUNTA

Califique la forma de guardar, almacenar o archivar la información y documentación en los almacenes “Multicomercio Centro”.

TABLA N° 5

OPCIONES	RESPUESTA	PORCENTAJE
Excelente	0	0,00%
Muy Bueno	0	0,00%
Bueno	1	100,00%
Regular	0	0,00%
Insuficiente	0	0,00%
Sin Respuesta	0	0,00%
TOTAL	1	100,00%



Fuente: Encuesta al Administrador de los almacenes “Multicomercio Centro”

Elaboración: Autores del proyecto de Tesis.

En virtud de que el Administrador de los Almacenes “Multicomercio Centro”, es una sola persona, se ha procedido a realizar la respectiva entrevista, obteniendo una respuesta buena para la primera pregunta, equivaliendo un porcentaje del 100% del criterio del administrador, determinándose que:

- La manera de almacenar o archivar la información que se maneja en los almacenes “Multicomercio Centro”, no es precisamente la más indicada y apropiada.
- Existe redundancia en el almacenamiento de los datos.
- No existe almacenada toda la información necesaria y requerida de los clientes.
- La documentación que se archiva es vulnerable de ser extraviada y en un eventual siniestro de perderse.

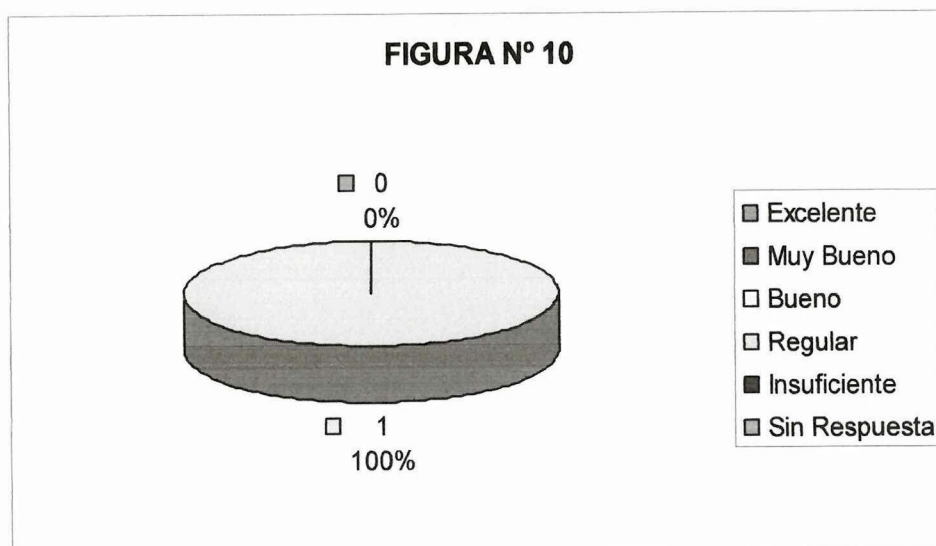
SEGUNDA PREGUNTA

Señale la manera de llevar el registro o inventario de los productos o artículos disponibles o en stock para su comercialización.

TABLA N° 6

OPCIONES	RESPUESTA	PORCENTAJE
Excelente	0	0,00%
Muy Bueno	0	0,00%
Bueno	0	0,00%
Regular	1	100,00%
Insuficiente	0	0,00%
Sin Respuesta	0	0,00%
TOTAL	1	100,00%

FIGURA N° 10



Fuente: Encuesta al Administrador de los almacenes “Multicomercio Centro”

Elaboración: Autores del proyecto de Tesis.

El Administrador de los Almacenes Multicomercio “Centro”, indica que la forma de llevar el registro o inventario de los productos o artículos disponibles o en stock para su comercialización, es regular expresado en el 100%, determinándose que:

- La forma manual en la que se mantiene el registro de los diferentes productos o artículos disponibles en stock para ser comercializados, no es la adecuada.
- Existe falta de agilidad para acceder y tener reportes de los artículos o productos en stock.

El análisis de resultados completo y en detalle se encuentra en el CD adjunto a la presente tesis, en la carpeta con el nombre **ANEXO II**, donde se puede apreciar claramente cada una de las preguntas formuladas a todos quienes intervienen directa o indirectamente en los almacenes “Multicomercio Centro”, con la representación gráfica y explícita de los resultados.

Cabe señalar además que la numeración de la tesis se altera, debido a que el resto del análisis de resultados se encuentra anexado en un archivo aparte, por lo tanto la numeración de ese archivo no es tomado en cuenta en la numeración general de la tesis.

3.2. CRITERIOS DE LOS CLIENTES.

El criterio de los clientes de los almacenes “Multicomercio Centro”, es el más importante porque la imagen y prestigio y por ende el engrandecimiento de los locales comerciales, depende en gran parte de la conformidad en la calidad de los productos y artículos y una excelente entrega de la información y atención que se brinde, fruto de la encuesta se detalla a continuación los siguientes resultados:

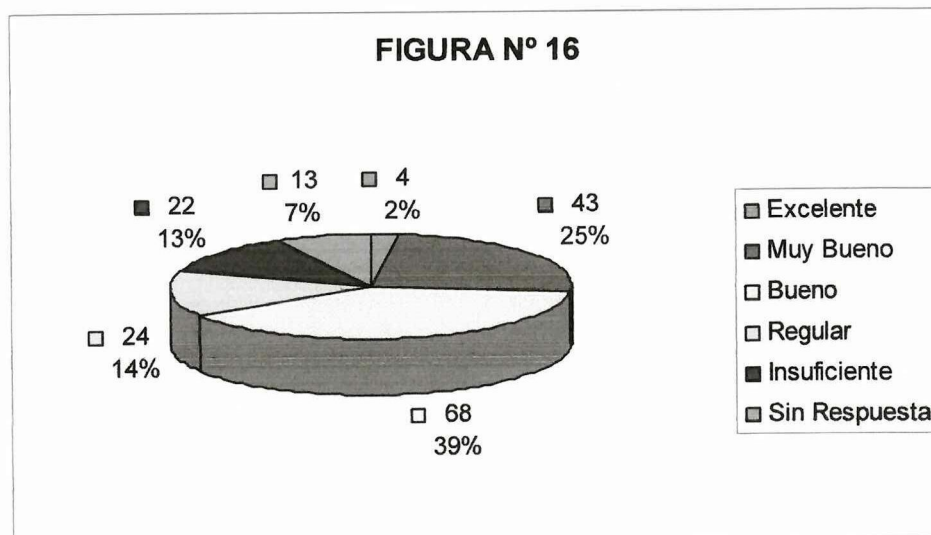
PRIMERA PREGUNTA

¿Cómo evaluaría la rapidez en la atención que le brinda el administrador de los almacenes “Multicomercio Centro”?

TABLA N° 12

OPCIONES	RESPUESTA	PORCENTAJE
Excelente	4	2,30%
Muy Bueno	43	24,71%
Bueno	68	39,08%
Regular	24	13,79%
Insuficiente	22	12,64%
Sin Respuesta	13	7,47%
TOTAL	174	100,00%

FIGURA N° 16



Fuente: Encuesta a los clientes de los almacenes “Multicomercio Centro”

Elaboración: Autores del proyecto de Tesis.

Siendo indagados los clientes sobre la rapidez y eficiencia con la que les atiende el administrador de los almacenes “Multicomercio Centro”, se ha obtenido: 4 clientes que corresponde al 2,30% opina que es excelente, 43 clientes que corresponden al 24,71% opina que es muy buena, 68 clientes que responden al 39,08% dice que es buena, 24 clientes correspondientes al 13,79% indican que es regular, 22 clientes que representan al 12,64% señalan insuficiente, y 13 clientes correspondiendo al 7,47% de la población no responde al cuestionamiento. De acuerdo a la opinión de la mayoría de la población de clientes, se concluye que la atención en la entrega de información por parte del Administrador es buena debido a:

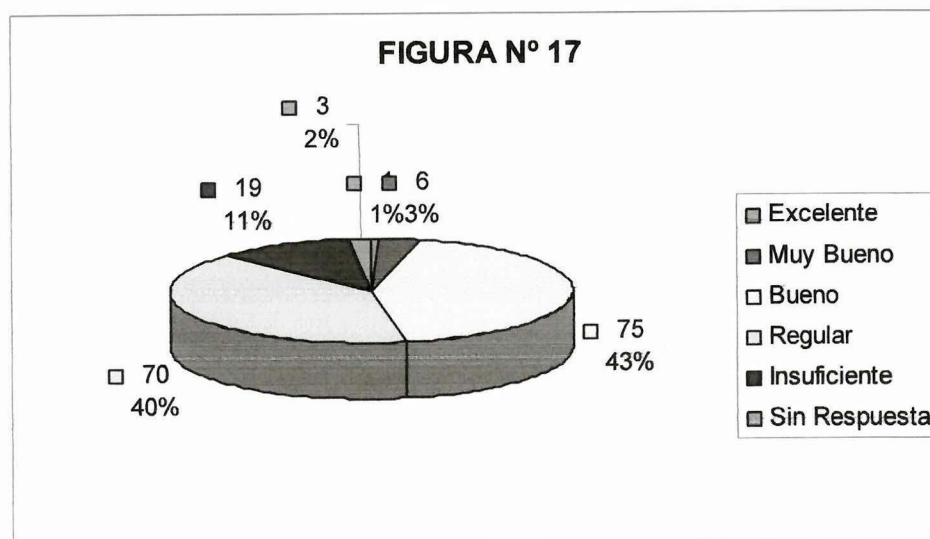
- La información de artículos y productos es almacenada en forma manual por lo que existe demora para acceder a la verificación de la misma.
- La información de los clientes se encuentra duplicada y desordenada lo que demora su acceso y difusión.
- Los cálculos de saldos y precios son manuales, por lo que tarda hasta efectuar los cálculos respectivos para su emisión.

SEGUNDA PREGUNTA

¿Cómo considera que es la agilidad en la entrega de información sobre los diferentes artículos y productos que están en exhibición en los almacenes “Multicomercio Centro”?

TABLA N° 13

OPCIONES	RESPUESTA	PORCENTAJE
Excelente	1	0,57%
Muy Bueno	6	3,45%
Bueno	75	43,10%
Regular	70	40,23%
Insuficiente	19	10,92%
Sin Respuesta	3	1,72%
TOTAL	174	100,00%



Fuente: Encuesta a los clientes de los almacenes “Multicomercio Centro”

Elaboración: Autores del proyecto de Tesis.

Con relación a la pregunta sobre cómo es la agilidad en la entrega de información sobre los diferentes artículos y productos que están en exhibición en los almacenes “Multicomercio Centro”, se obtiene lo siguiente: 1 cliente que corresponde al 0,57% opina que es excelente, 6 clientes que corresponden al 3,45% opinan que es muy buena, 75 clientes que responden al 43,10% dicen que es buena, 70 clientes correspondientes al 40,23% indican que es regular, 19 clientes que representan al 10,92% señalan que es insuficiente, y 3 clientes correspondiendo al 1,72% de la población se abstienen. De acuerdo a la opinión de la mayoría de la población de clientes, se determina que la entrega de información a los clientes es buena debido a:

- La información de artículos y productos es almacenada en forma manual por lo que existe demora para acceder a la verificación de la misma.

El análisis de resultados completo y en detalle se encuentra en el CD adjunto a la presente tesis, en la carpeta con el nombre **ANEXO II**, donde se puede apreciar claramente cada una de las preguntas formuladas a todos quienes intervienen directa o indirectamente en los almacenes “Multicomercio Centro”, con la representación gráfica y explícita de los resultados.

Cabe señalar además que la numeración de la tesis se altera, debido a que el resto del análisis de resultados se encuentra anexado en un archivo aparte, por lo tanto la numeración de ese archivo no es tomado en cuenta en la numeración general de la tesis.

3.3. CRITERIOS DE LOS PROVEEDORES.

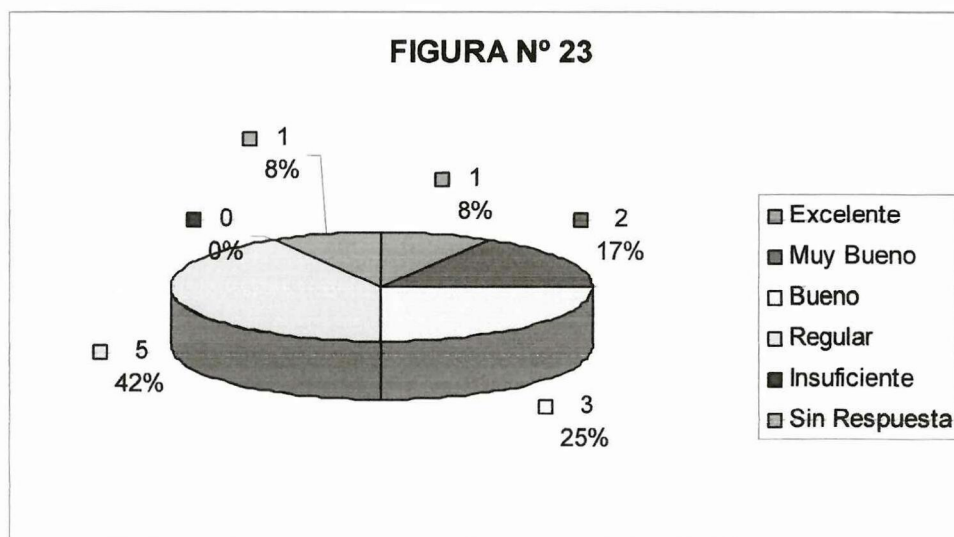
El criterio de los proveedores de los almacenes “Multicomercio Centro”, es también importante, ya que es con quienes se mantiene relaciones comerciales de forma permanente, relaciones de pedidos de artículos o productos en función de stock o existencias, solicitudes de créditos a corto y largo plazo, emisión de registros de ventas de sus respectivos artículos.

PRIMERA PREGUNTA

¿Cómo calificaría la agilidad en la revisión de la información sobre las existencias o stock de los productos que ofrece los almacenes “Multicomercio Centro”?

TABLA N° 19

OPCIONES	RESPUESTA	PORCENTAJE
Excelente	1	8,33%
Muy Bueno	2	16,67%
Bueno	3	25,00%
Regular	5	41,67%
Insuficiente	0	0,00%
Sin Respuesta	1	8,33%
TOTAL	12	100,00%



Fuente: Encuesta a los proveedores de los almacenes “Multicomercio Centro”

Elaboración: Autores del proyecto de Tesis.

De acuerdo al cuestionamiento: ¿Cómo calificaría la agilidad en la revisión de la información sobre las existencias o stock de los productos que ofrece los almacenes “Multicomercio Centro”?; se obtiene los siguientes datos: 1 proveedor que corresponde al 8,33% opina que es excelente, 2 proveedores que corresponden al 16,67% opinan que es muy buena, 3 proveedores que responden al 25,00% dicen que es buena, 5 proveedores correspondientes al 41,67% indican que es regular, ningún proveedor señala que es insuficiente, y 1 proveedor correspondiente al 8,33% de la población no expresa su opinión. De lo señalado por la mayoría de la población de proveedores, se desprende que:

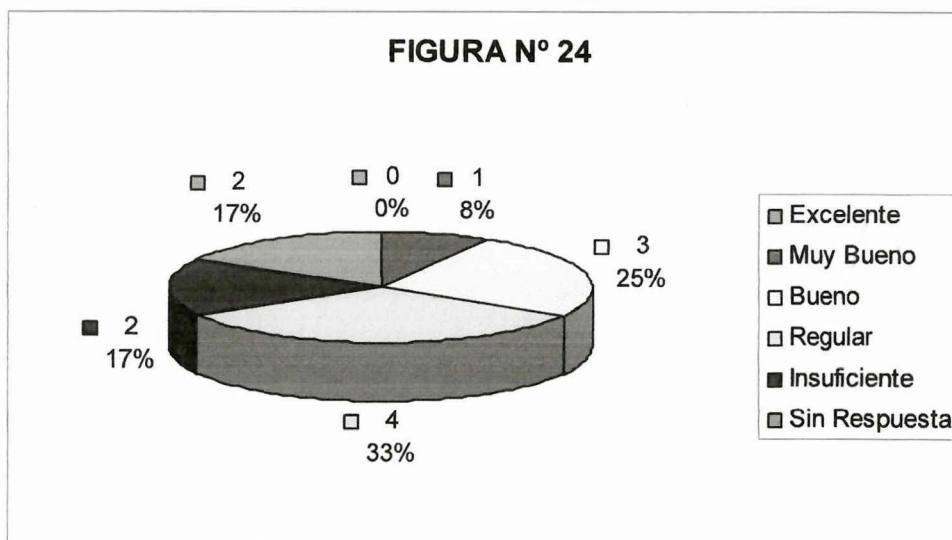
- Existe lentitud y demora al momento realizar la revisión de la información sobre las existencias o stock de los productos, puesto que se las efectúa de forma manual y se lleva o mantiene en forma documentada.

SEGUNDA PREGUNTA

¿Cómo calificaría el proceso de ingreso de información para solicitar un pedido?

TABLA N° 20

OPCIONES	RESPUESTA	PORCENTAJE
Excelente	0	0,00%
Muy Bueno	1	8,33%
Bueno	3	25,00%
Regular	4	33,33%
Insuficiente	2	16,67%
Sin Respuesta	2	16,67%
TOTAL	12	100,00%



Fuente: Encuesta a los proveedores de los almacenes “Multicomercio Centro”

Elaboración: Autores del proyecto de Tesis.

En relación a la pregunta: ¿Cómo calificaría el proceso de ingreso de información para solicitar un pedido?; se obtienen los siguientes datos: ningún proveedor opina que es excelente, 1 proveedor que corresponde al 8,33% opinan que es muy bueno, 3 proveedores que responden al 25,00% dicen que es bueno, 4 proveedores correspondientes al 33,33% indican que es regular, 2 proveedores representando al 16,67% señalan que es insuficiente, y 2 proveedores correspondientes al 16,67% de la población no expresa su opinión. De lo señalado por la mayoría de la población de proveedores, se desprende que:

- El proceso manual de ingreso de información para solicitar un pedido no es el apropiado, ya que demora el pedido y por ende la entrega de artículos y productos.

El análisis de resultados completo y en detalle se encuentra en el CD adjunto a la presente tesis, en la carpeta con el nombre **ANEXO II**, donde se puede apreciar claramente cada una de las preguntas formuladas a todos quienes intervienen directa o indirectamente en los almacenes “Multicomercio Centro”, con la representación gráfica y explícita de los resultados.

Cabe señalar además que la numeración de la tesis se altera, debido a que el resto del análisis de resultados se encuentra anexado en un archivo aparte, por lo tanto la numeración de ese archivo no es tomado en cuenta en la numeración general de la tesis.

3.5. CRITERIOS DE LOS VENDEDORES.

Para realizar la presente investigación se ha desarrollado una encuesta dirigida a cada uno de los recursos humanos que intervienen en el funcionamiento de los almacenes “Multicomercio Centro”, de la ciudad de Latacunga, en esta sección se verán los criterios de los vendedores en función a cada uno de los cuestionamientos formulados por los integrantes del proyecto de tesis.

La población total de vendedores para el caso es de 4 personas.

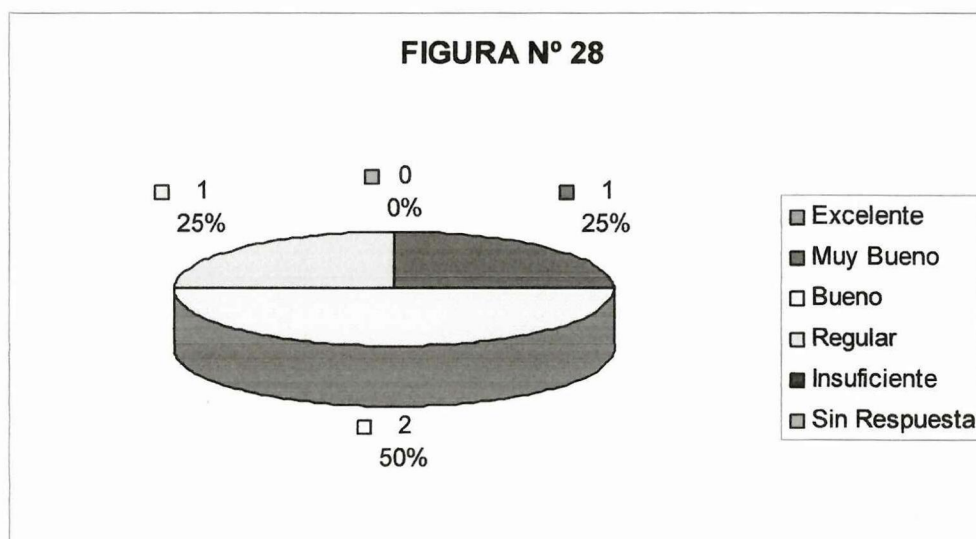
Los vendedores serán quienes se encuentren en constante relación e interacción con el sistema, puesto manejan la mayoría de transacciones.

PRIMERA PREGUNTA

¿De que manera define la forma como se almacena la información sobre productos y artículos?

TABLA N° 24

OPCIONES	RESPUESTA	PORCENTAJE
Excelente	0	0,00%
Muy Bueno	1	25,00%
Bueno	2	50,00%
Regular	1	25,00%
Insuficiente	0	0,00%
Sin Respuesta	0	0,00%
TOTAL	4	100,00%



Fuente: Encuesta a los vendedores de los almacenes “Multicomercio Centro”

Elaboración: Autores del proyecto de Tesis.

Como resultados a la pregunta: ¿De que manera define la forma como se almacena la información sobre productos y artículos?; se obtienen los siguientes datos: ningún vendedor opina que es excelente, 1 vendedor que corresponde al 25,00% opina que es muy bueno, 2 vendedores que responden al 50,00% señalan que es bueno, 1 vendedor correspondiente al 25,00% indica que es regular, ninguna vendedor señala insuficiente, y todos expresan su opinión. De lo indicado por la mayoría de la población de vendedores, se desprende que:

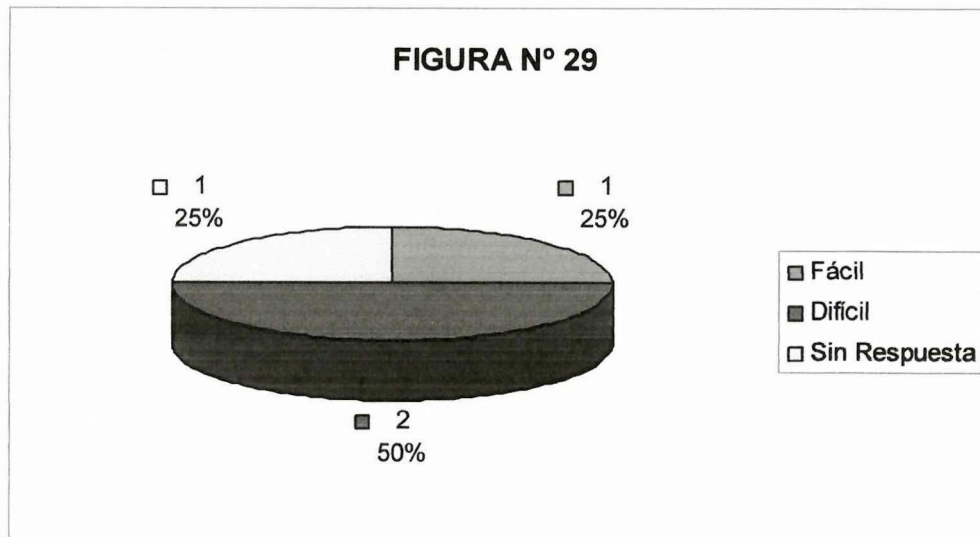
- La forma en la que se lleva o almacena la información sobre productos y artículos al interior de la los almacenes “Multicomercio Centro”, es buena, pero por ser llenada o ingresada de manera manual y al ser llevada en forma de registros o documentos es vulnerable o volátil a ser violentada o extraviada, y al acceder o requerir la misma demora los procesos.

SEGUNDA PREGUNTA

¿Indique el grado de facilidad o dificultad en la búsqueda de información de los clientes?

TABLA N° 25

OPCIONES	RESPUESTA	PORCENTAJE
Fácil	1	25,00%
Difícil	2	50,00%
Sin Respuesta	1	25,00%
TOTAL	4	100,00%



Fuente: Encuesta a los vendedores de los almacenes “Multicomercio Centro”

Elaboración: Autores del proyecto de Tesis.

Como resultados a la pregunta: ¿Indique el grado de facilidad o dificultad en la búsqueda de información de los clientes?; se obtienen los siguientes datos: 1 vendedor que corresponde al 25,00% opina que es Fácil, 2 vendedores que responden al 50,00% señalan que es Difícil, 1 vendedor correspondiente al 25,00% no da su criterio. De lo respondido por la mayoría de la población de vendedores, se deduce que:

- Existe un cierto grado de dificultad para los vendedores al momento de realizar la búsqueda de la información de los diferentes artículos o productos, para ser dada a conocer a los clientes, lo cual hace que exista demora en la entrega de la información requerida por el cliente.

El análisis de resultados completo y en detalle se encuentra en el CD adjunto a la presente tesis, en la carpeta con el nombre **ANEXO II**, donde se puede apreciar claramente cada una de las preguntas formuladas a todos quienes intervienen directa o indirectamente en los almacenes “Multicomercio Centro”, con la representación gráfica y explícita de los resultados.

Cabe señalar además que la numeración de la tesis se altera, debido a que el resto del análisis de resultados se encuentra anexado en un archivo aparte, por lo tanto la numeración de ese archivo no es tomado en cuenta en la numeración general de la tesis.

3.5. ANÁLISIS EXPERIENCIAS SIMILARES DE OTROS LOCALES

COMERCIALES AFINES.

Se ha realizado un análisis de locales comerciales similares, donde se puede constatar que no existe un adecuado sistema informático que permita realizar actividades de control, administración y gestión de la información, que son los requerimientos básicos de una entidad comercial.

El software que se utiliza en algunas de las casas comerciales de la ciudad de Latacunga, no se apegan a las necesidades de las mismas, sino que éstas deben ajustarse a las utilidades del sistema informático que manejan.

De acuerdo a este análisis tenemos una base concreta para el diseño e implementación de un sistema multiusuario para el control, administración y gestión de la información para los almacenes “Multicomercio Centro”, el cuál será desarrollado de acuerdo a las necesidades y exigencias de la casa comercial, pudiendo ser manipulado sin mucho esfuerzo, brindando eficiencia y confianza en el manejo de la información.

CAPÍTULO

IV

PROPUESTA: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA MULTIUSUARIO PARA EL CONTROL, ADMINISTRACIÓN Y GESTIÓN DE LA INFORMACIÓN PARA LOS ALMACENES “MULTICOMERCIO CENTRO” DE LA CIUDAD DE LATACUNGA.

4.1. Propuesta

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA MULTIUSUARIO PARA EL CONTROL, ADMINISTRACIÓN Y GESTIÓN DE LA INFORMACIÓN PARA LOS ALMACENES “MULTICOMERCIO CENTRO” DE LA CIUDAD DE LATACUNGA.

4.1.1. Modelo de Fases para el desarrollo del software

4.1.1.1. Análisis

4.1.1.1.1. Fase de Planificación y Especificación de Requisitos

El sistema “SISCAYGIMC_1.0”, se lo realizará con la notación del Lenguaje de Modelamiento Unificado (UML - Unified Modeling Language) porque es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reutilizables.

UML es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar de facto de la industria, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh. Estos autores

fueron contratados por la empresa Rational Software Co. para crear una notación unificada en la que basar la construcción de sus herramientas CASE. En el proceso de creación de UML han participado, no obstante, otras empresas de gran peso en la industria como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupos de analistas y desarrolladores.

4.1.1.1.1.1. Planificación

4.1.1.1.1.1.1. Propósito

En esta fase se busca conocer y describir de una forma general el ámbito del sistema, alcance, usuarios, funciones y sus requisitos, los mismos que se relacionan con los procesos de manejo de la información de que se realizan en los almacenes “Multicomercio Centro” de la ciudad de Latacunga.

4.1.1.1.1.1.2. Ámbito del Sistema

El estudio de factibilidad realizado al administrador, empleados, proveedores y clientes, demuestra que es necesario el desarrollo e implementación de un software fiable que permita obtener una información rápida y eficiente, mediante el uso de herramientas de última generación, permitiendo tener una mejor gestión y control de los datos, brindando un apoyo informático incondicional e invaluable a los almacenes “Multicomercio Centro”.

4.1.1.1.1.1.3. Alcance del Sistema

El alcance del sistema “SISCAYGIMC_1.0”, ha sido determinado por la población de clientes, empleados, proveedores y administrador que pertenecen a los almacenes “Multicomercio Centro”, detallándose:

Tabla 32. Población general de los almacenes “Multicomercio Centro”

DENOMINACIÓN PERSONAL	POBLACIÓN
Clientes	500
Proveedores	12
Vendedores	4
Administradores	1
Total población:	517

Fuente: Los autores.

4.1.1.1.1.4. Definición de Usuarios

Tabla 33. Usuarios del Sistema

USUARIO	ACTIVIDAD
Administrador	Se encarga de la administración total del Sistema
Vendedores	Gestionan las actividades que se cumplen de acuerdo a las potestades dadas por el administrador.
Proveedores	Solicita la realización de una operación al Sistema.
Clientes	Solicita la realización de una operación al Sistema.

Fuente: Los autores.

4.1.1.1.2. Descripción General

Se trata de describir todas las funciones, características y limitantes que se deben tomar en cuenta en el desarrollo del sistema

“SISCAYGIMC_1.0”, con el fin de dar a conocer las generalidades del mismo.

4.1.1.1.2.1. Funciones del Sistema

El análisis del flujo de las actividades que habitualmente se realizan en los almacenes “Multicomercio Centro”, determina que el sistema “SISCAYGIMC_1.0”, deberá contener las siguientes funciones:

- Gestión de Clientes
- Gestión de Empleados
- Gestión de Proveedores
- Gestión de Productos y Artículos
- Gestión de Facturación

Los procesos que cumplen cada una de las funciones antes mencionadas se detallan en la siguiente tabla:

Tabla 34. Gestión de Procesos

GESTIÓN	PROCESOS			
	ALTAS	BAJAS	CAMBIOS	REPORTES
Clientes	√		√	√
Empleados	√		√	√
Proveedores	√		√	√
Productos y artículos	√	√	√	√
Facturación			√	√

Fuente: Los autores.

4.1.1.1.3. Requisitos Específicos

4.1.1.1.3.1. Requisitos Funcionales

4.1.1.1.3.1.1. Gestión de Clientes

El sistema “SISCAYGIMC_1.0”, podrá realizar los siguientes procesos:

- Ingreso de nuevos clientes.
- Modificación de los datos de los clientes.
- Reporte de clientes por Apellido o por número de cédula.

4.1.1.1.3.1.2. Gestión de Empleados

El sistema “SISCAYGIMC_1.0”, podrá realizar los siguientes procesos:

- Ingreso de nuevos empleados.
- Modificación de los datos de los empleados.
- Reporte de empleados por Apellido.

4.1.1.1.3.1.3. Gestión de Proveedores

El sistema “SISCAYGIMC_1.0”, podrá realizar los siguientes procesos:

- Ingreso de nuevos proveedores.
- Modificación de los datos de los proveedores.
- Reporte de proveedores por Razón Social.

4.1.1.1.3.1.4. Gestión de Productos y Artículos

El sistema “SISCAYGIMC_1.0”, podrá realizar los siguientes procesos:

- Ingreso de nuevos productos y artículos.
- Eliminación de productos y artículos.
- Modificación de los datos de los productos y artículos.
- Reporte de productos y artículos por nombre.
- Reporte de productos y artículos por marca.

4.1.1.1.3.1.5. Gestión de Facturación

El sistema “SISCAYGIMC_1.0”, podrá realizar los siguientes procesos:

- Modificación de los datos de una factura (anulación de factura).
- Reporte de las facturas emitidas por cliente.
- Reporte de las facturas emitidas por fecha.

4.1.1.1.3.2. Requisitos de Rendimiento

El sistema “SISCAYGIMC_1.0”, mantendrá un óptimo rendimiento, ya que será desarrollado con un conjunto de herramientas de última tecnología como son C#.NET y SQL Server 2000 que soportan grandes flujos de información y ofrecen el entorno cliente servidor para el trabajo en una Intranet.

4.1.1.1.3.3. Requisitos Tecnológicos

4.1.1.1.3.3.1. Requisitos de Hardware

Para el óptimo funcionamiento del sistema “SISCAYGIMC_1.0”, es necesario el siguiente hardware:

- Un computador servidor que contemple las siguientes características:

Procesador: Pentium IV de 800 Mhz o superior

Memoria RAM: 256 MB o superior

Disco Duro: 80 Gb o superior

Tarjeta de red: 10/100 o superior

- Computadores clientes que contemplen las siguientes características:

Procesador: Pentium III de 866 Mhz o superior

Memoria RAM: 128 MB o superior

Disco Duro: 30 Gb o superior

Tarjeta de red: 10/100 o superior

- Un concentrador (Switch o HUB) de 4 puertos o superior

4.1.1.1.3.3.2. Requisitos de Software

Servidor

- Sistema Operativo Windows XP Profesional, Windows 2000 Server o Windows 2003 Server.
- Visual Studio .NET 2003
- SQL Server 2000
- Internet Information Service 5.5 (IIS)
- Internet Explorer 6.0

Clientes

- Sistema Operativo Windows 95 o superior.
- Framework 1.1 Visual Studio .NET 2003
- Internet Explorer 4.0 o superior

4.1.1.1.3.4. Requisitos de Desarrollo

El sistema “SISCAYGIMC_1.0”, presenta las particularidades necesarias para ser desarrollado empleando el modelo de fases, permitiendo crear un estándar de software.

4.1.1.1.3.5. Requisitos de Comunicación

Se utilizará la conexión cliente-servidor de los computadores existentes en los almacenes “Multicomercio Centro”, por medio de la configuración de red y se procederá a utilizar uno de ellos como servidor para la instalación del software.

4.1.1.1.3.6. Atributos

4.1.1.1.3.6.1. Seguridad

El tipo de seguridad que empleará el sistema “SISCAYGIMC_1.0”, para su interacción, se detalla bajo los siguientes aspectos:

- Verificación del ingreso del número de cédula por medio de un algoritmo de autenticación.
- Restricciones a páginas webs para el ingreso a opciones de altas, bajas y cambios, mediante el uso de cuentas de usuario con su respectiva contraseña que está encriptada en la base de datos.
- Restricciones a páginas de ingreso, eliminación y modificación mediante Cookies.

4.1.1.1.3.6.2. Tipos de Usuarios

4.1.1.1.3.6.2.1. Administrador del Sistema

Persona que tiene el acceso total al sistema y es la encargada de gestionar y controlar todos los procesos efectuados con la información ingresada en la base de datos.

4.1.1.1.3.6.2.2. Empleados

Personas encargadas de realizar todas las actividades que se efectúan en los almacenes “Multicomercio Centro”, estando en relación directa con el sistema pudiendo realizar altas y cambios.

4.1.1.1.3.6.2.3. Clientes

Son aquellas personas que pueden consultar o buscar información referente a su estado de crédito, saldos, productos, precios, novedades e información en general.

4.1.1.1.3.6.2.4. Proveedores

Personas o empresas que pueden consultar información referente a productos, artículos, stock, etc.

4.1.1.2. Diseño

Esta fase consiste en realizar los pasos propios del diseño de en una aplicación de gestión clásica, manteniendo al margen el hecho de que

existan necesidades de navegación e interfaz abstracta. En el diseño hay que seleccionar la arquitectura que mejor soporte al sistema, así como los casos de uso que se hayan diseñado en fases anteriores del proceso de desarrollo. Además, habrá que obtener el modelo de clases de diseño.

El diseño es una parte central de todas las actividades que conducen a la producción del software. Diseñamos para:

- Para comunicar la estructura deseada y el comportamiento de nuestro sistema.
- Para visualizar y controlar la arquitectura del sistema.
- Para comprender mejor el sistema que estamos construyendo, muchas veces descubriendo oportunidades para la simplificación y la reutilización.
- Para controlar el riesgo.

El diseño requiere del análisis exhaustivo de los requerimientos y proporciona un modelo que destaca los planos del sistema. Los modelos pueden involucrar planos detallados, así como planos más generales que ofrecen una visión global del sistema en consideración. Un buen modelo incluye aquellos elementos menores que no son relevantes para el nivel de abstracción dado. Todo sistema puede ser descrito desde diferentes perspectivas utilizando diferentes modelos, y cada modelo es por tanto una abstracción semánticamente cerrada del sistema. Un modelo puede ser estructural, destacando la organización del sistema, o puede ser de comportamiento, resaltando su dinámica.

4.1.1.2.1. Diagrama de Casos de Uso

El diagrama de casos de uso representa la forma en como un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso).

Un diagrama de casos de uso (*diagrama use-case*), puede ser dividido en varios diagramas de caso de uso. Un diagrama de casos de uso contiene elementos de modelo para el sistema, los actores y los casos de uso y muestra las diferentes relaciones tales como generalización, asociación y dependencia entre estos elementos. El diagrama de caso de uso debe ser fácil de entender por el usuario final

Los elementos de un diagrama de casos de uso son:

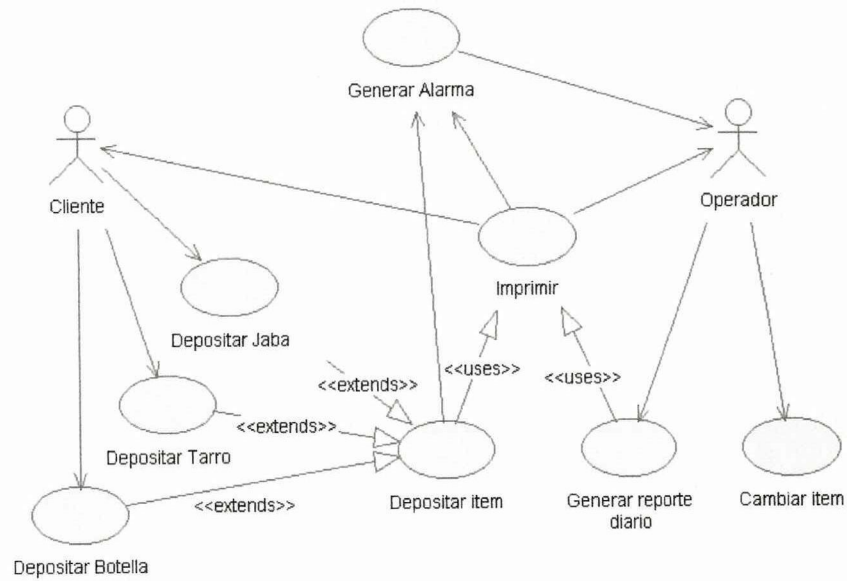
- **Actores:** Un actor es algo con comportamiento, como una persona (identificada por un rol), un sistema informatizado u organización, y que realiza algún tipo de interacción con el sistema. Se representa mediante una figura humana dibujada con palotes. Esta representación sirve tanto para actores que son personas como para otro tipo de actores.
- **Casos de Uso:** Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad, y se representa en el

Diagrama de Casos de Uso mediante una elipse con el nombre del caso de uso en su interior. El nombre del caso de uso debe reflejar la tarea específica que el actor desea llevar a cabo usando el sistema.

- **Relaciones entre Casos de Uso:** Un caso de uso, en principio, debería describir una tarea que tiene un sentido completo para el usuario. Sin embargo, hay ocasiones en las que es útil describir una interacción con un alcance menor como caso de uso. La razón para utilizar estos casos de uso no completos en algunos casos, es mejorar la comunicación en el equipo de desarrollo, el manejo de la documentación de casos de uso. Para utilizar estos casos de uso más pequeños, las relaciones entre estos y los casos de uso ordinarios pueden ser de los siguientes tres tipos:
 - **Incluye (<>):** Un caso de uso base incorpora explícitamente a otro caso de uso en un lugar especificado en dicho caso base. Se suele utilizar para encapsular un comportamiento parcial común a varios casos de uso.
 - **Extiende (<>):** Cuando un caso de uso base tiene ciertos puntos (puntos de extensión) en los cuales, dependiendo de ciertos criterios, se va a realizar una interacción adicional. El caso de uso que extiende describe un comportamiento opcional del sistema.

- **Generalización ():** Cuando un caso de uso definido de forma abstracta se particulariza por medio de otro caso de uso más específico. Se representa por una línea continua entre los dos casos de uso, con el triángulo que simboliza generalización en UML (usado también para denotar la herencia entre clases) pegado al extremo del caso de uso más general. Al igual que en la herencia entre clases, el caso de uso hijo hereda las asociaciones y características del caso de uso padre. El caso de uso padre se trata de un caso de uso abstracto, que no está definido completamente.

Figura 36. Diagrama de Casos

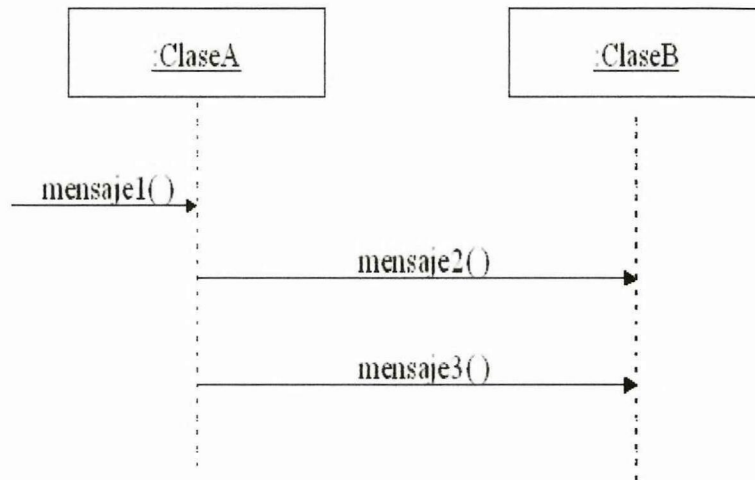


Fuente: <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/index.html>

4.1.1.2.2. Diagrama de Secuencias.

Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo. El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba abajo. Se pueden colocar etiquetas (como restricciones de tiempo, descripciones de acciones, etc.) bien en el margen izquierdo o bien junto a las transiciones o activaciones a las que se refieren.

Figura 37. Diagrama de Secuencias.



Fuente: <http://www.clikear.com/manuales/uml/>

4.1.1.2.3. Diagrama de Clases

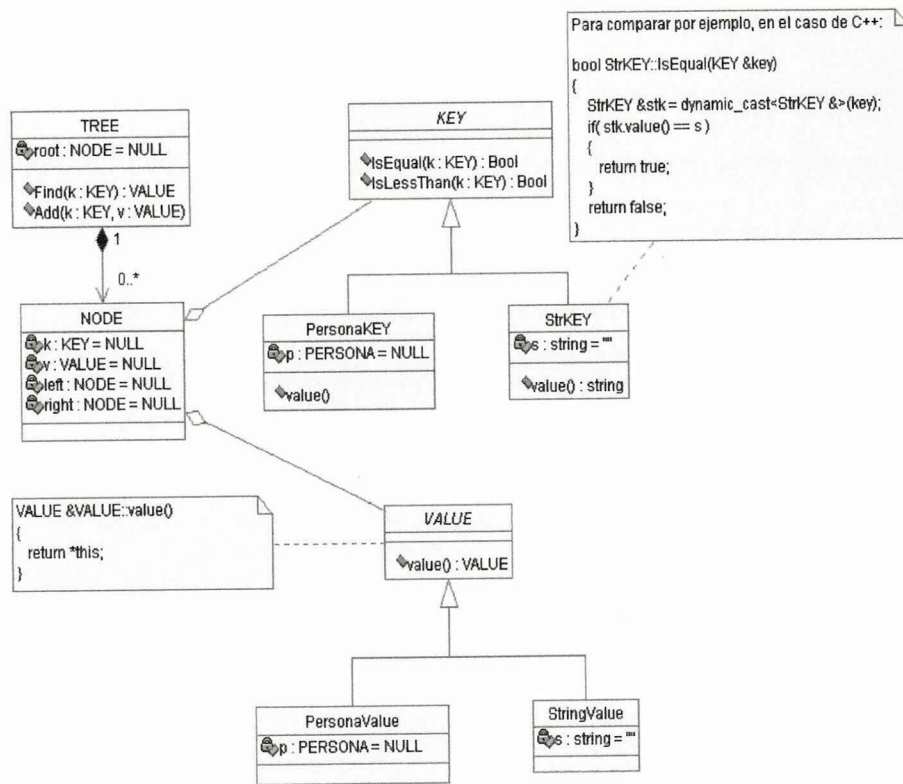
El Diagrama de Clase es el diagrama principal de diseño y análisis para un sistema. En él, la estructura de clases del sistema se especifica, con relaciones entre clases y estructuras de herencia. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal. Durante el diseño, se usa el mismo diagrama, y se modifica para satisfacer los detalles de las implementaciones.

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenimiento.

Un diagrama de clases esta compuesto por los siguientes elementos:

- **Clase:** atributos, métodos y visibilidad.
- **Relaciones:** Herencia, Composición, Agregación, Asociación y Uso.

Figura 38. Diagrama de Clases



Fuente: <http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html>

4.1.1.2.4. Diseño Lógico

El diseño lógico traduce los escenarios de uso creados en el diseño conceptual en un conjunto de objetos de negocio y sus servicios. El diseño lógico se convierte en parte en la especificación funcional que se usa en el diseño físico. El diseño lógico es independiente de la tecnología. El diseño lógico refina, organiza y detalla la solución de negocios y define formalmente las reglas y políticas específicas de negocios.

Un objeto de negocios es la encapsulación de un servicio que abstrae las cualidades esenciales de algo de interés.

Un servicio es una unidad con capacidad de cómputo. Un servicio debe satisfacer lo siguiente:

- Ser seguro, lo que equivale a un uso correcto y con autorización.
- Ser válido, qué tareas o reglas se pueden aplicar.
- Manejar excepciones, informando al cliente.
- Contar con un catálogo de servicios que constituye un repositorio de servicios.

Los objetos de negocio deben verificarse y probarse de tal manera que asegure que los módulos operen como unidades completas de trabajo.

Las tareas de verificación incluyen:

- Una verificación independiente:
 - Pre y post condiciones.
 - Lógica y funcionalidad individual.
- Una verificación dependiente:
 - Verificación de dependencias.
 - Que operan como una unidad específica de trabajo.

El diseño lógico comprende las siguientes tareas:

- Identificar y definir los objetos de negocio y sus servicios.
- Definir las interfases.
- Identificar las dependencias entre objetos.
- Validar contra los escenarios de uso.

- Comparar con la arquitectura de la empresa.
- Revisar y refinar tanto como sea necesario.

Para definir los objetos de negocios y sus servicios se puede usar la técnica de análisis nombre-verbo de los escenarios de uso. También se puede emplear la técnica sujeto-verbo-objeto directo. En estas técnicas los sujetos y el objeto directo son los candidatos a objetos de negocio y los verbos activos son los candidatos a servicios.

Una interfase tiene las siguientes partes:

- Nombre.
- Precondiciones, lo que debe estar presente antes de ejecutarse.
- Postcondiciones, estado final.
- Capacidad o funcionalidad (SQL, pseudocódigo, función matemática).
- Dependencias.

La tarea de identificar las dependencias entre objetos permite identificar eventos, sucesos o condiciones que permitan la realización de tareas de negocios coordinadamente o transaccionalmente. Para ello se debe considerar lo siguiente:

- Identificar los eventos disparadores (triggers).
- Determinar cualquier dependencia (existencial o funcional).
- Determinar cualquier problema de consistencia o secuencia.
- Identificar cualquier regulación de tiempo crítica.

- Considerar algún problema organizacional (transacciones).
- Identificar y auditar los requerimientos de control.
- Determinar lugares y dependencias a través de la ubicación.
- Determinar cuando el servicio que controla la transacción es dependiente de los servicios contenidos en otros objetos de negocio.

La validación del modelo lógico debe ser tal que éste sea:

- Completo – debe representar todos los escenarios de uso,
- Correcto – el comportamiento lógico debe corresponder con el comportamiento conceptual, y
- Claro – los objetos de negocio y servicios no deben ser ambiguos

4.1.1.2.5. Diseño Físico

El diseño físico traduce el diseño lógico en una solución implementable y costo-efectiva o económica.

El componente es la unidad de construcción elemental del diseño físico.

Las características de un componente son:

- Se define según cómo interactúa con otros.
- Encapsula sus funciones y sus datos.
- Es reusable a través de las aplicaciones.
- Puede verse como una caja negra.
- Puede contener otros componentes.

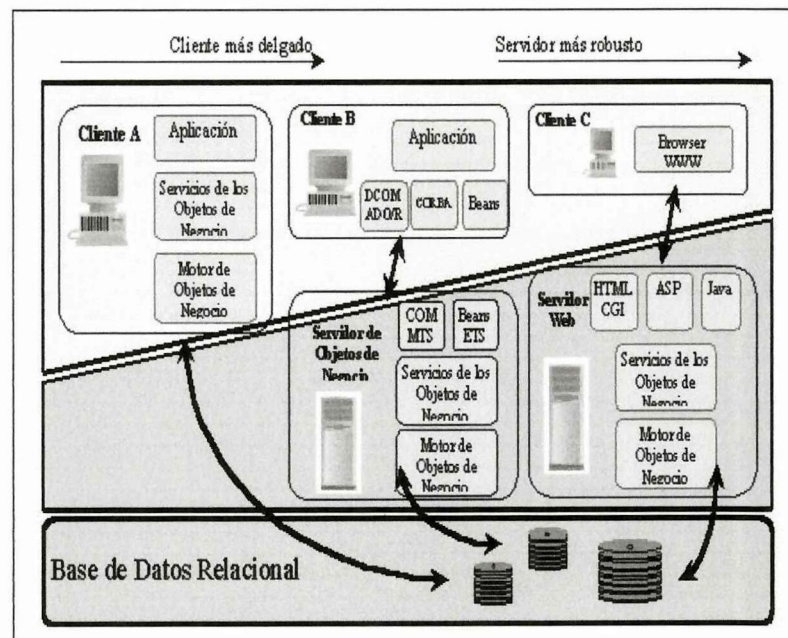
En el diseño físico se debe cuidar el nivel de granularidad (un componente puede ser tan grande o tan pequeño según su funcionalidad, es decir, del tamaño tal que pueda proveer de una funcionalidad compleja pero de control genérico) y la agregación y contención (un componente puede reusar utilizando técnicas de agregación y contención, sin duplicar código).

El diseño físico debe involucrar:

- El diseño para distribución – debe minimizarse la cantidad de datos que pasan como parámetros entre los componentes y éstos deben enviarse de manera segura por la red.

- El diseño para multitarea – debe diseñarse en términos de la administración concurrente de dos o más tareas distintas por una computadora y el *multithreading* o múltiples hilos de un mismo proceso).
- El diseño para uso concurrente – el desempeño de un componente remoto depende de si está corriendo mientras recibe una solicitud.
- El diseño con el manejo de errores y prueba de eventos.

Figura 39. *Arquitectura física de tres capas de la aplicación cliente/servidor.*



Fuente: <http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html>

El diseño físico comprende las siguientes tareas:

- Definir los componentes.
- Refinar el empaquetamiento y distribución de componentes.
- Especificar las interfases de los componentes.
- Distribuir los componentes en la red.

- Distribuir los repositorios físicos de datos.
- Examinar la tolerancia a fallas y la recuperación de errores.
- Validar el diseño físico.

De las tareas anteriores la más importante es la distribución de los datos que pueden ser centralizados, una partición, un extracto o una réplica.

Los datos centralizados equivalen a una base de datos maestra ubicada en un lugar central. No hay copias de los datos.

Una partición de datos es una segmentación de la base de datos maestra. Es útil cuando los datos se pueden fragmentar fácilmente y actualizarse en un sitio local con cambios frecuentes. No hay superposición entre particiones. En una partición horizontal cada hilera existe en una sola base de datos. En una partición vertical cada columna es contenida en una y solo una base de datos.

Un extracto de datos es una copia de toda o una porción de la base de datos maestra. No se permite la actualización. Se usa un *timestamp* o etiqueta de tiempo para indicar qué tan viejos son los datos.

Una réplica de datos es un fragmento de la bases de datos maestra que se puede actualizar. Una réplica de datos es cuando el sitio de actualización cambia a un sitio local. No se permiten actualizaciones en la base de datos réplica y en la base de datos maestra a la vez, por lo que debe de haber sincronización entre ambas.

El diseño físico está íntimamente ligado a una alternativa tecnológica. Ante la acelerada evolución tecnológica es importante considerar los estándares del momento y las tendencias ya que una mala decisión implicará un costo enorme (en dinero y en tiempo) al actualizarse a otra plataforma distinta.

La tendencia actual en la arquitectura cliente/servidor es crear el back-end como un servidor robusto multitareas y *multithreading* y el front-end como un cliente muy delgado que no acapare al servidor comunicándose entre sí en una plataforma internet con protocolos estándar en redes heterogéneas.

4.1.1.2.6. Diccionario de Datos

Contiene las características lógicas de los sitios donde se almacenan los datos del sistema, incluyendo nombre, descripción, alias, contenido y organización. Identifica los procesos donde se emplean los datos y los sitios donde se necesita el acceso inmediato a la información, se desarrolla durante el análisis de flujo de datos y auxilia a los analistas que participan en la determinación de los requerimientos del sistema, su contenido también se emplea durante el diseño.

Razones para su utilización:

1. Para manejar los detalles en sistemas muy grandes, ya que tienen enormes cantidades de datos, aun en los sistemas mas chicos hay gran cantidad de datos.

Los sistemas al sufrir cambios continuos, es muy dificil manejar todos los detalles. Por eso se registra la información, ya sea sobre hoja de papel o usando procesadores de texto. Los analistas mas organizados usan el diccionario de datos automatizados diseñados específicamente para el análisis y diseño de software.

2. Para asignarle un solo significado a cada uno de los elementos y actividades del sistema.

Los diccionarios de datos proporcionan asistencia para asegurar significados comunes para los elementos y actividades del sistema y registrando detalles adicionales relacionados con el flujo de datos en el sistema, de tal manera que todo pueda localizarse con rapidez.

3. Para documentar las características del sistema, incluyendo partes o componentes así como los aspectos que los distinguen. También es necesario saber bajo que circunstancias se lleva a cabo cada proceso y con que frecuencia ocurren. Produciendo una comprensión más completa. Una vez que las características están articuladas y

registradas, todos los participantes en el proyecto tendrán una fuente común de información con respecto al sistema.

4. Para facilitar el análisis de los detalles con la finalidad de evaluar las características y determinar donde efectuar cambios en el sistema.

Determina si son necesarias nuevas características o si están en orden los cambios de cualquier tipo.

Se abordan las características:

- Naturaleza de las transacciones: las actividades de la empresa que se llevan a cabo mientras se emplea el sistema.
 - Preguntas: solicitudes para la recuperación o procesamiento de información para generar una respuesta específica.
 - Archivos y bases de datos: detalles de las transacciones y registros maestros que son de interés para la organización.
 - Capacidad del sistema: Habilidad del sistema para aceptar, procesar y almacenar transacciones y datos.
5. Localizar errores y omisiones en el sistema, detectan dificultades, y las presentan en un informe. Aun en los manuales, se revelan errores.

Contenido de un registro del diccionario

El diccionario tiene dos tipos de descripciones para el flujo de datos del sistema, son los elementos datos y estructura de datos.

Elemento dato: son los bloques básicos para todos los demás datos del sistema, por si mismos no le dan un significado suficiente al usuario. Se agrupan para formar una estructura de datos.

Descripción: Cada entrada en el diccionario consiste de un conjunto de detalles que describen los datos utilizados o producidos por el sistema, cada uno esta identificado con:

- *Un nombre:* para distinguir un dato de otro.
- *Descripción:* indica lo que representa en el sistema.
- *Alias:* porque un dato puede recibir varios nombres, dependiendo de quien uso este dato.
- *Longitud:* porque es de importancia de saber la cantidad de espacio necesario para cada dato.
- *Valores de los datos:* porque en algunos procesos solo son permitidos valores muy específicos para los datos. Si los valores de los datos están restringidos a un intervalo específico, esto debe estar en la entrada del diccionario.

Estructura de datos: es un grupo de datos que están relacionados con otros y que en conjunto describen un componente del sistema.

4.1.1.3. Codificación.

Dado que los proyectos de sistemas de información se diseñan teniendo en mente ahorro de espacio, costo y tiempo, los sistemas de codificación en los cuales las condiciones, palabras ideas o relaciones se expresan a través de un código, se desarrollan para reducir errores de control y acelerar el proceso en su totalidad. Un código es un número breve, título o símbolo utilizado en vez de descripciones más largas o ambiguas. Cuando ocurre un suceso los detalles del mismo se resumen a menudo mediante un código. Son necesarios menos detalles en la entrada y no se tienen como resultado ninguna pérdida de información. Los tipos de métodos de codificación:

4.1.1.3.1. Código Clasificación

Los Código Clasificación colocan entidades separadas, como hechos, personas y objetos en grupos distintos llamados clases. Un código se utiliza para distinguir una clase de otra. El usuario registra el código en un documento fuente o puede teclearlo directamente al sistema a través de la terminal en un sistema en línea. El usuario (Ya sea que haya aprendido los

códigos o los consulte) clasifica el hecho en una de varias de varias categorías posibles y registra el código.

4.1.1.3.2. Código Función

Los códigos función establecen las actividades o el trabajo que han de llevar a cabo sin tener que anotar todos los detalles en forma descriptiva.

Los analistas utilizan este tipo de código en forma frecuente en los datos de transacción para señalar al sistema como procesar los mismos.

4.1.1.3.3. Código Secuencia

Los códigos secuenciales son números o letras asignados en serie. Señalan el orden en el cual han ocurrido los hechos: por ejemplo un sistema bancario debe ser capaz de mantener un seguimiento del orden de las transacciones, de manera que sea claro que transacción procesar primero, cual después y así sucesivamente; por lo tanto se debe especificar el numero de secuencia en el diseño con el propósito de ordenar las transacciones.

Los códigos secuenciales son utilizados también con el propósito de identificación, pero se asignan en el orden en el cual los clientes entran al sistema.

4.1.1.3.4. Código Dimuto Significativo

Un esquema de codificación bien concebido que utilice subcódigos dentro de códigos más grandes o números, puede proporcionar una buena información a los usuarios. Supóngase que los números de artículos se asignan a los diferentes materiales y productos que una compañía almacena o vende. Una forma es asignar números en secuencia, comenzando por el primero y continuando hasta el último, o pueden añadir un prefijo para los números de identificación, de manera que pueda describirse más adelante el tipo de artículo: Acero tiene un prefijo "A", plástico uno "P".

Los códigos pueden dividirse en subconjuntos o subcódigos, caracteres que son parte del número de identificación que tienen un significado especial. Los subcódigos indican al usuario información adicional sobre el artículo, Ejemplo La clase de producto, el proveedor, esta información se lleva a través de dígitos significativos.

4.1.1.3.5. Código Nemo-técnico

Los Código Nemo-técnico utilizan letras y símbolos del producto para describirlo en una forma que comunique visualmente. Por ejemplo; para describir un televisor a color de 21", un código útil es TV-CL-21", (en blanco y negro sería TV-BW-21"). Es difícil confundir el **nemo-técnico** TV con el de otro producto, las universidades siempre utilizan nemo-técnico para codificar a los estudiantes: Msc. Master en ciencias, Dra. Doctores en ciencias.

La codificación de los datos y las transacciones reduce la cantidad de datos que se necesita introducir y simplifican el proceso, de manera que se reduce la probabilidad de errores. La selección de un código sobre otro dependerá por su puesto, de los datos y de los objetivos del analista.

4.1.1.4. Pruebas del Sistema

Se efectuaron dos tipos de pruebas con el sistema; la primera prueba funcional se hizo para comprobar la funcionalidad del “SISCAYGIMC_1.0”. Esta prueba se la realizó ingresando un grupo de datos de comprobación, la misma incluyó la validación de campos, reglas de negocio, integración de procesos e ingreso de datos. Además la prueba se la realizó tanto en el cliente así como en el servidor, comprobando su correcta funcionalidad en la arquitectura Cliente/Servidor.

La siguiente prueba se la realizó mediante la generación de reportes usando como parámetros los datos de los clientes y datos de los productos o artículos adquiridos para el ejemplo.

A continuación se muestra las pruebas efectuadas en el sistema “SISCAYGIMC_1.0”:

Para ingresar al sistema tenemos dos formas de acceso, la una es pública; es decir que cualquier tipo de usuario puede ingresar a la página principal y efectuar transacciones o gestiones públicas en lo referente al almacén

comercial como: visitar el sitio de reseña e información general, ofertas de productos, cotizar precios de los diferentes productos o artículos.

La otra forma de acceso al “SISCAYGIMC_1.0” es restringida para el administrador del sistema y usuarios mediante el ingreso del nombre de usuario y su respectiva contraseña para hacer uso e interactuar con el resto del sistema.

4.1.1.4.1. Prueba de Datos

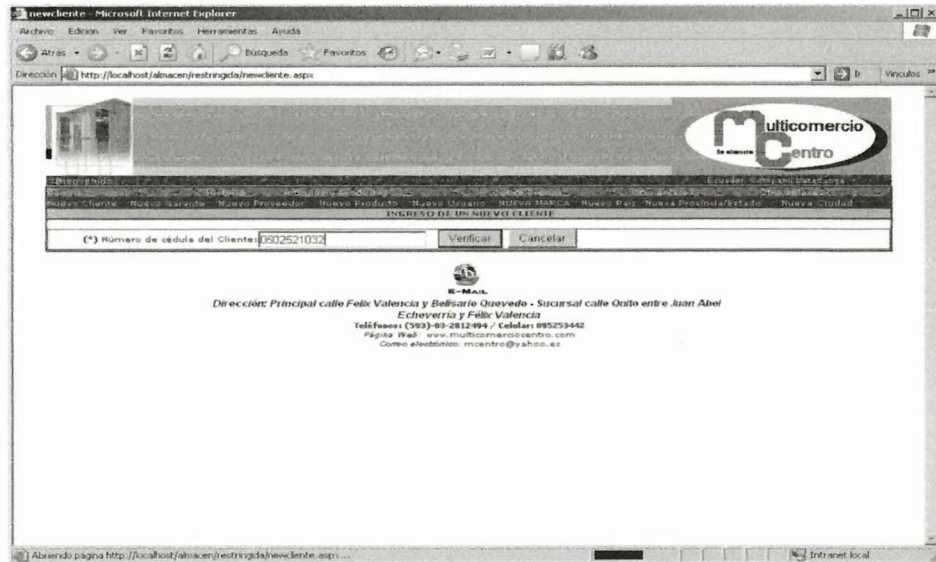
Esta prueba consiste en comprobar si el usuario ingresó correctamente los campos requeridos para el almacenamiento en los respectivos campos dentro de la base de datos; como se muestra a continuación:

Tabla 35. Prueba de Datos

ACCIÓN	DATO DE PRUEBA	RESULTADO
Ingresar el número de cédula del cliente.	0502521032	Ingreso de datos generales del cliente.
Comprobar el número de cédula de clientes existentes.	0502521032	Estado del cliente con el detalle general del producto o artículo adquirido y estado del crédito.
Cambio de clave del administrador.	Clave anterior + clave nueva + rescribir clave nueva	Se ha cambiado la clave correctamente.
Lista de productos o artículos.	tipo	Listado de productos o artículos según el tipo seleccionado.
Nomina de clientes por fecha.	fecha	Nomina de clientes correspondiente a la o las fechas solicitadas.

Fuente: Los Autores.

Figura 40. Ingreso del número de cédula (Página de ingreso de nuevo cliente)



Fuente: Los Autores

4.1.1.4.2. Prueba de Validaciones

Esta prueba consiste en verificar que los datos ingresados cumplan con las diferentes normativas con la utilización de algunos algoritmos; como por ejemplo para el caso de la verificación de la cédula de identidad del cliente, ingreso de campos principales o mandatarios como nombres, apellidos, etc. de acuerdo a la estructura de la base de datos.

Tabla 36. Prueba de Validaciones

ACCIÓN	DATO DE PRUEBA	RESULTADO
Ingreso de cédula de identidad del cliente.	0502085970	Cédula válida.
Marque el género del cliente.	(opción no marcada)	¡Error! Seleccione un género del cliente.
Selección del estado civil del cliente.	(campo no seleccionado)	¡Error! Seleccione un estado civil correcto.
Ingreso de correo electrónico.	pruebacorreo@yahoo	¡Error! Verifique si ingresó correctamente el

		e-mail.
Ingreso de nombre del cliente.	(campo vacío)	Debe ingresar el nombre del cliente.
Nuevo cliente.	0502085970	El cliente que desea ingresar, ya existe en la base de datos de "Multicomercio Centro".

Fuente: Los Autores.

Figura 41. Errores al ingresar datos de un cliente

INGRESO DE DATOS DE UN NUEVO CLIENTE

(*) Apellidos del Cliente: Debe ingresar los apellidos!

(*) Nombres del Cliente: Debe ingresar los nombres!

(*) Género del Cliente: Masculino Femenino Debe marcar el género!

(*) Estado civil del Cliente:

Seleccione el País del Cliente: Actualizar Región

Provincia del Cliente:

(*) Seleccione la ciudad del Cliente:

(*) Dirección del domicilio del Cliente: Debe ingresar la dirección del domicilio!

Teléfono del domicilio del Cliente:

(*) Profesión u ocupación del Cliente:

(*) Dirección del lugar de trabajo del Cliente: Debe ingresar la dirección del lugar de trabajo!

Teléfono del lugar de trabajo del Cliente:

Correo electrónico del Cliente:

Observación especial del Cliente:

Nota: Los campos marcados por (*) son obligatorios de ser llenados, ingresados o seleccionados.

Fuente: Los Autores

4.1.1.4.3. Prueba de Seguridades

- **Autenticación en ASP.NET [Formularios (cookie)]:**

El proveedor de autenticación por formularios es un esquema de autenticación que permite a la aplicación recopilar credenciales utilizando

un formulario HTML directamente del cliente. El cliente envía las credenciales directamente al código de la aplicación para la autenticación. Si la aplicación autentica al cliente, genera una cookie que el cliente presenta en solicitudes posteriores. Si una solicitud de un recurso protegido no contiene la cookie, la aplicación redirige al cliente a la página de inicio de la sesión. (Ver CD anexo **Manual de Usuario**).

Figura 42. Autenticación en ASP.NET (Página principal)



Fuente: Los Autores

Tabla 37. Prueba de Seguridad

ACCIÓN	DATO DE PRUEBA	RESULTADO
Sistema inactivo durante 10 minutos en cualquier parte del mismo.	Tiempo transcurrido	Redireccionamiento a la página principal.

Fuente: Los Autores

- **Encriptación de claves por medio del algoritmo de HASH**

Es conveniente la utilización del algoritmo de Hash para la seguridad de la clave de acceso a todo el sistema “SISCAYGIMC_1.0”.

El proceso con el cual trabaja el algoritmo de Hash, es calcular un valor computacionalmente irreversible para cada clave y almacenar dicho valor en lugar de la clave ingresada por el usuario, en su forma original. Para confirmar que se ha ingresado la clave correcta simplemente se aplica el cálculo explicado anteriormente al valor ingresado y se compara los dos resultados (el recién ingresado y el almacenado), si coinciden se concluye que el valor original era el mismo.

El término “Computacionalmente irreversible” quiere decir que no se puede regresar desde el valor calculado hasta la clave original aún usando super-computadoras durante largo tiempo. Los algoritmos que hacen este tipo de cálculo son conocidos desde hace tiempo y se conocen colectivamente como “hashes”. Los dos algoritmos más populares de este tipo tienen el nombre de SHA1 y MD5.

Figura 43. Prueba de Seguridades [Encriptación algoritmo Hash (SHA1)]

Data in Table 'USUARIO' in 'BDAlmacen' on '(local)						
CEDULA_USR	NOMBRE_USR	APELLIDO_USR	TIPO_USR	LLAVE_USR	OBS1_USR	OBS2_USR
0502523111	r	h	Vendedor	23F8316D3183C8388894342A		
VENDEDOR1	STEVE	GUACHO	VENDEDOR	1C78323B456946A0376F9263D	<NULL>	<NULL>

Fuente: Los Autores

Tabla 38. Prueba de Seguridades [Encriptación algoritmo Hash (SHA1)]

ACCIÓN	DATO DE PRUEBA	ENCRIPCIÓN
Ingreso de la clave del Administrador.	*****	123F8316D3183C838B894342A

Fuente: Los Autores

4.1.1.5. Fase de Mantenimiento del Sistema

El mantenimiento del sistema “SISCAYGIMC_1.0”, se lo efectuará de acuerdo a las necesidades y exigencias de los requerimientos de los usuarios y de las operaciones que vayan siendo requeridas en el transcurso de su uso y operación.

Por ejemplo la necesidad de crear un módulo para el ingreso de nuevos países, provincias o estados y ciudades o localidades, en el caso de extenderse internacionalmente la empresa comercial.

Dentro del mantenimiento del sistema, es importante destacar la necesidad de respaldar los datos e información creando copias de seguridad, tanto del sistema mismo, como de la base de datos, a fin de evitar posibles pérdidas de datos e información y si los hubiere tener a mano respaldo actualizados frecuentemente para poder enfrentar una posible emergencia.

También es conveniente tomar en cuenta el ciclo de vida del software, para lo cual se ha estimado una durabilidad de siete años y sujeto a reformas se lo

ajustaría mediante módulos a fin de cumplir tareas y requerimientos adicionales y que surjan en el transcurso del tiempo.

4.1.2. Elaboración del Manual del Usuario del Sistema

El manual del usuario contiene datos referentes al correcto y adecuado uso del sistema “SISCAYGIMC_1.0”, a éste se debe recurrir al producirse algún error o al existir alguna inquietud en cuanto a la interacción con este software. El manual del usuario en detalle se encuentra adjunto a la tesis en material magnético (CD), en la carpeta con el nombre ANEXO 4.

4.1.3. Elaboración del Manual del Programador del Sistema

El manual del programador del sistema está compuesto por datos relacionados con la programación fuente, siendo tomadas para el ejemplo determinadas líneas de código realizado por los desarrolladores de éste proyecto. El manual del programador en detalle se encuentra adjunto a la tesis en material magnético (CD), en la carpeta con el nombre ANEXO 5.

4.1.4. Implementación del Sistema

Dentro de la implementación del sistema, primero hay que asegurarse que el mismo sea funcional y operacional para que los usuarios puedan manipular e interactuar de forma sencilla, correcta, ordenada y rápida con el sistema “SISCAYGIMC_1.0”.

El sistema “SISCAYGIMC_1.0”, es una aplicación cliente/servidor por lo tanto necesita estar interconectado a través de una Intranet y cumplir de forma estricta con los requerimientos mínimos ya detallados anteriormente, tanto de Hardware para un buen rendimiento del equipo en su uso, como de Software especialmente en el computador que haga las veces de servidor, ya que deberá soportar o alojar el motor de base de datos que para el caso es SQL Server, así como también alojará el sistema en su naturaleza misma, el cual para su ejecución y funcionamiento requiere de la instalación previa del Framework SDK v1.1 e Internet Information Service (IIS), como se detalla en el manual del programador que se encuentra adjunto en el CD.

4.1.5. Ciclo de vida del Sistema

El ciclo de vida del “SISCAYGIMC_1.0” se lo considera tomando en cuenta un modelo clásico del desarrollo de sistemas, dentro del cuál ya se ha desarrollado y detallado las fases de análisis, requerimientos, diseño, desarrollo e implementación, las cuales resultan efectivas si son aplicadas de manera adecuada para el efecto y dentro del tiempo establecido.

CAPÍTULO

V

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES.

- El correcto uso de las técnicas de investigación conjuntamente con las técnicas de programación y desarrollo de software, permitieron dar solución al problema planteado, contribuyendo de ésta manera al desarrollo tecnológico de los almacenes “Multicomercio Centro” de la ciudad de Latacunga.
- La implementación del software “SISCAYGIMC_1.0”, cumple con los requerimientos establecidos por la casa comercial y está acorde a las necesidades de la misma, permitiendo brindar una mejor y eficiente atención a sus clientes.
- El software “SISCAYGIMC_1.0”, por sus características permite la implementación de nuevos módulos, si así lo requiere la empresa comercial, así mismo, consta de una fácil interconectividad con la Internet, para brindar a los usuarios una rápida y eficiente navegación en la red.
- El software “SISCAYGIMC_1.0”, puede implementarse sin ninguna dificultad en otras casas comerciales que requieran de un sistema de ésta naturaleza, para el control y gestión de la información.

5.2. COMPROBACIÓN DE LA HIPÓTESIS PLANTEADA.

“La falta de un sistema de control, administración y gestión de la información en los almacenes “Multicomercio Centro”, ha conllevado a una falencia en la utilización de los recursos tecnológicos de la empresa comercial.”

En base a los resultados de las encuestas aplicadas a todas las personas que intervienen directa e indirectamente en tal empresa comercial, se hace necesario implementar nuevas herramientas tecnológicas actuales para la gestión de las actividades, mismas que permitan un mejor control y manejo de la información de la empresa comercial y por ende brindar un mejor servicio a los clientes y proveedores, generando una mejor administración de los almacenes “Multicomercio Centro”.

El sistema multiusuario para el control, administración y gestión de la información “SISCAYGIMC_1.0”, permite incorporar tecnologías informáticas de última generación en la empresa comercial, concluyendo que la hipótesis es verdadera.

5.3. RECOMENDACIONES.

- Es necesario la capacitación a todo el personal de los almacenes “Multicomercio Centro”, con el fin de dar a conocer las ventajas y la forma de uso del sistema desarrollado, para su posterior buen funcionamiento y resultados útiles finales.
- Se recomienda recurrir a los manuales el usuario y/o programador, según sea el requerimiento para despejar inquietudes o aclarar posibles errores, durante la interacción con el sistema.

- Es importante tomar en cuenta los requerimientos tanto de software como de hardware, para un correcto funcionamiento del sistema, así como también es menester tener actualizado el antivirus del equipo de cómputo para evitar posibles daños parciales o totales del software.
- El presente sistema por su estructura y fácil manejo, es recomendable utilizarlo en cualquier otra casa comercial que lo requiera, contribuyendo de esta forma al progreso del sector comercial de la provincia.
- Se recomienda respaldar la información ingresada periódicamente, con el objeto de evitar posibles pérdidas de datos que pueden ocurrir por cualquier motivo de fuerza mayor.

5.4. GLOSARIO DE TÉRMINOS.

A

ADO.NET.- Conjunto de tecnologías de acceso a datos incluidas en las bibliotecas de clases de .NET Framework que proporcionan acceso a datos relacionales y a XML. ADO.NET se compone de clases que forman el DataSet (como tablas, filas, columnas, relaciones, etcétera), de proveedores de datos de .NET Framework y de definiciones de tipos personalizados (como SqlTypes para SQL Server).

Aplicación Web ASP.NET (ASP.NET Web application).- Aplicación que procesa solicitudes HTTP (solicitudes Web) y se ejecuta en ASP.NET. Una aplicación Web de ASP.NET puede incluir páginas ASP.NET, servicios Web XML, controladores HTTP y módulos HTTP. Conjunto de tecnologías de Microsoft .NET Framework para crear aplicaciones Web y servicios Web XML.

Adaptador de dispositivo (device adapter).- Clase de marco de trabajo de página ASP.NET que adapta el comportamiento de los controles y las páginas móviles tomando como base el dispositivo de destino.

Autenticación (authentication).- En la seguridad de .NET Framework, es el proceso de descubrir y comprobar la identidad de un principal mediante el examen de las credenciales del usuario y su consulta a una autoridad determinada.

Autorización (authorization).- En la seguridad de .NET Framework, se trata del proceso de limitar derechos de acceso mediante la concesión o negación de permisos específicos a una identidad autenticada o un principal.

ASCII.- American Standard Code of Information Exchange. Estándar mundial de 128 caracteres, letras, números y símbolos aceptados por la mayoría de editores y procesadores de texto.

Archivo de código subyacente (code-behind file).- Archivo de código que contiene la clase de página que implementa la lógica de programa de una aplicación de formularios Web Forms o de formularios Web Forms móviles de ASP.NET.

Archivo ejecutable (executable file).- Archivo con el formato ejecutable portable (PE) que el cargador del sistema operativo puede cargar en memoria y ejecutar. Puede ser un archivo .exe o .dll. En el contexto de .NET, Common Language Runtime debe traducir un archivo PE a código nativo para que el sistema operativo pueda ejecutarlo. Vea también: PE, archivo ejecutable portable.

Atributo (attribute).- Declaración descriptiva que anota elementos de programación como tipos, campos, métodos y propiedades. Los atributos se guardan con los metadatos de un archivo de .NET Framework y pueden utilizarse para describir el código para Common Language Runtime o para influir en el comportamiento de la aplicación en tiempo de ejecución.

Autorización basada en funciones (role-based authorization).- En la seguridad de .NET Framework, proceso de determinar si se permite a un principal realizar o no una acción solicitada. La autorización tiene lugar después de la autenticación y utiliza

información acerca de la identidad del principal y su función asociada para determinar a qué recursos puede tener acceso un principal.

B

Base de Datos.- Conjunto de datos interrelacionales y estructurado, almacenados de forma que puedan servir para todos los programas que lo puedan utilizar.

BROWSER.- Navegador, programa que permite visualizar paginas web y acceder a otros servicios de Internet.

BITS.- Código de un número binario que consiste en dos valores: cero (0) y uno (1). También es la información que se puede almacenar en una celda sencilla de memoria.

BYTE.- Es un grupo de bits que tienen un significado singular. Un byte representa ocho bits.

BACK UP.- Respaldo o copia de seguridad. Consiste en salvar todo lo que tienes en el disco duro a un periférico o dispositivo externo, DAT, CDROM, otro disco Duro, Disco extraíble, Flash Memory, etc.

Biblioteca de clases de .NET Framework (.NET Framework class library).- Biblioteca compatible con CLS de clases, interfaces y tipos de valor incluidos en .NET Framework SDK. Esta biblioteca brinda acceso a la funcionalidad del sistema y es la base sobre la que se crean las aplicaciones, los componentes y los controles de .NET Framework.

Base de aplicación (application base).- Directorio donde se encuentra el archivo .exe que se carga en el dominio de aplicación inicial o predeterminado.

C

C# .- Nuevo lenguaje de programación diseñado para crear aplicaciones empresariales que se ejecutan en .NET Framework. C#, que es una evolución de C y C++, garantiza la seguridad de tipos y está orientado a objetos. Puesto que se compila como código administrado, aprovecha los servicios de Common Language Runtime, como interoperabilidad de lenguaje, mayor seguridad y recolección de elementos no utilizados.

Common Language Runtime.- Motor que es el núcleo de la ejecución de código administrado. El motor de tiempo de ejecución proporciona al código administrado servicios como integración entre varios lenguajes, seguridad de acceso a código, administración de la duración de los objetos, y compatibilidad con la depuración y la generación de perfiles.

Common Language Specification (CLS).- Subconjunto de funciones del lenguaje admitidas por Common Language Runtime, incluyendo funciones comunes de varios lenguajes de programación orientados a objetos. Se garantiza que las herramientas y los componentes compatibles con CLS pueden interoperar con otras herramientas y componentes compatibles con CLS.

Common Object File Format (COFF).- En programación de 32 bits, formato de archivos ejecutables (imagen) y de objeto que puede transportarse a distintas

plataformas. La implementación de Microsoft se denomina formato de archivo ejecutable portable (PE).

Control de servidor Web (Web server control).- Control de servidor ASP.NET que pertenece al espacio de nombres System.Web.UI.WebControls. Los controles de servidor Web son más ricos y más abstractos que los controles de servidor HTML. Un control de servidor Web tiene un prefijo de etiqueta asp, como `<asp:Button runat="server" />`, en una página ASP.NET.

Control de usuario (user control).- En ASP.NET: control de servidor creado de manera declarativa utilizando la misma sintaxis que una página ASP.NET y guardado como archivo de texto con la extensión .ascx. Los controles de usuario permiten dividir y reutilizar la funcionalidad de página. Al solicitarlo, el marco de trabajo de páginas analiza un control de usuario en una clase derivada de System.Web.UI.UserControl y compila dicha clase en un ensamblado que vuelve a utilizar en solicitudes posteriores. Los controles de usuario son fáciles de desarrollar debido a su creación de estilos de página y a su implementación sin previa compilación.

En formularios Windows Forms: control compuesto que proporciona un comportamiento y una interfaz de usuario coherentes dentro de una misma aplicación o en diversas aplicaciones. El control de usuario puede ser local para una aplicación, o se puede agregar a una biblioteca y compilarlo en un archivo DLL para que lo utilicen múltiples aplicaciones.

Código administrado (managed code).- Código ejecutado por el entorno de Common Language Runtime en lugar de ejecutarlo directamente el sistema operativo. Las aplicaciones de código administrado obtienen servicios de Common Language Runtime, como la recolección automática de elementos no utilizados, la comprobación del tipo de motor de tiempo de ejecución y la compatibilidad con la seguridad, entre otros. Estos servicios proporcionan un comportamiento uniforme independiente de la plataforma y del lenguaje de las aplicaciones de código administrado.

Código nativo (native code).- Código que se ha compilado en código máquina específico del procesador.

Código no administrado (unmanaged code).- Código ejecutado directamente por el sistema operativo, fuera del entorno de Common Language Runtime. El código no administrado debe suministrar sus servicios propios de recolección de elementos no utilizados, comprobación de tipos, compatibilidad con la seguridad, etc., al contrario que el código administrado, que recibe estos servicios de Common Language Runtime.

Compatible con CLS (CLS-compliant).- Código que expone públicamente sólo las funciones de lenguaje incluidas en la especificación Common Language Specification. La compatibilidad con CLS puede aplicarse a clases, interfaces, componentes y herramientas.

Compilación Just-In-Time (JIT compilation).- Compilación que convierte el Lenguaje intermedio de Microsoft (MSIL) en código máquina cuando se requiere el código en tiempo de ejecución.

Comprobación (verification).- Parte del proceso de compilación en la que se comprueba que el código cumple un determinado conjunto de reglas definidas como prueba de unos requisitos de seguridad específicos. Common Language Runtime puede comprobar el Lenguaje intermedio de Microsoft (MSIL).

CD-ROM.- Compact Disc-Read Only Memory. Aplicación de la tecnología digital y láser a la informática, aunque también se utilizan en grabaciones musicales.

Campo (field).- Miembro que representa una variable asociada con un objeto o una clase.

Clase (class).- Tipo de referencia que encapsula datos (constantes y campos) y el comportamiento (métodos, propiedades, indizadores, eventos, operadores, constructores de instancia, constructores estáticos y destructores), y puede contener tipos anidados. Los tipos de clase admiten la herencia, un mecanismo mediante el cual una clase derivada puede extender y especializar una clase base.

D

Datos administrados (managed data).- Objetos cuyas duraciones están administradas por Common Language Runtime. El motor de tiempo de ejecución controla automáticamente la disposición de los objetos y administra las referencias a éstos, liberándolos cuando ya no se utilizan.

Delegado (delegate).- Tipo de referencia que es la versión administrada de un puntero a función de C++. Los delegados pueden hacer referencia tanto a métodos de instancia

como a métodos estáticos (en Visual Basic, **Shared**), mientras que los punteros a función sólo pueden hacer referencia a métodos estáticos (en Visual Basic, **Shared**).

Denegación (denial).- Al denegar un permiso, un método puede impedir que los llamadores aprovechen el privilegio que representa dicho permiso. Si un método de la pila de llamadas deniega el permiso A, una comprobación del permiso A en el recorrido de pila producirá un error, a menos que se encuentre una aserción válida en la pila, entre el método que realiza la denegación y el método que inició la comprobación del permiso A.

Devolución (postback).- Proceso en el que una página Web devuelve datos a la misma página en el servidor.

Directiva de seguridad (security policy).- Directiva activa establecida por el administrador que genera mediante programación los permisos concedidos para todo el código administrado basándose en los permisos solicitados por el código. No se permite la ejecución del código que requiere más permisos de los que concederá la directiva.

Dominio de aplicación (AppDomain, application domain).- Límite que Common Language Runtime establece alrededor de los objetos creados dentro del mismo ámbito de aplicación (es decir, cualquier lugar de la secuencia de activaciones de objetos que empieza en el punto de entrada de la aplicación). Los dominios de aplicación ayudan a aislar los objetos creados en una aplicación de los creados en otras aplicaciones, de forma que se pueda predecir el comportamiento en tiempo de ejecución. En un único proceso pueden existir varios dominios de aplicación.

Duración (lifetime).- Período de tiempo que empieza cuando se asigna un objeto en memoria y termina cuando el recolector de elementos no utilizados elimina el objeto de la memoria.

E

E-mail.- Electronic mail Envío de mensajes electrónicos a otros usuarios de la red con dirección electrónica.

Ejecución simultánea (side-by-side execution).- Capacidad de instalar y utilizar múltiples versiones de un ensamblado de forma aislada y al mismo tiempo. La ejecución simultánea se puede aplicar a los componentes y a las aplicaciones y también a .NET Framework. Permitir que los ensamblados coexistan y se ejecuten simultáneamente en el mismo equipo es esencial para tener un control de versiones sólido en Common Language Runtime.

Encapsulación (encapsulation).- Posibilidad de que un objeto oculte sus datos y métodos internos, haciendo que sólo sean accesibles mediante programación las partes deseadas del objeto.

Enlace de datos (data binding).- Asociación de un origen de datos con un control de servidor.

Ensamblado (assembly).- Conjunto de uno o varios archivos que pertenecen a una versión y se implementan como unidad. Un ensamblado es la unidad de creación principal de una aplicación .NET Framework. Todos los tipos y recursos administrados

se incluyen en un ensamblado y se marcan como accesibles únicamente dentro del ensamblado o bien como accesibles desde código de otros ensamblados. Los ensamblados también juegan un papel clave en la seguridad. El sistema de seguridad de acceso a código utiliza información acerca del ensamblado para determinar el conjunto de permisos que se concede al código del ensamblado.

Espacio de nombres (namespace).- Esquema de nombres lógico para agrupar los tipos relacionados. .NET Framework utiliza un esquema de nombres jerárquico para agrupar los tipos en categorías lógicas de funcionalidad relacionada, como la tecnología ASP.NET o la funcionalidad de interacción remota. Las herramientas de diseño pueden utilizar espacios de nombres para que los programadores puedan examinar y hacer referencia más fácilmente a los tipos en el código. Un ensamblado individual puede contener tipos cuyos nombres jerárquicos tienen distintas raíces de espacio de nombres y una raíz de espacio de nombres lógico puede abarcar varios ensamblados. En .NET Framework, un espacio de nombres es una comodidad para la nomenclatura lógica en tiempo de diseño, mientras que un ensamblado establece el ámbito de nombres para los tipos en tiempo de ejecución.

Estado de vista (view state).- Configuración de propiedades actual de una página ASP.NET y de todos los controles de servidor ASP.NET contenidos en la página. ASP.NET puede detectar cuándo se solicita un formulario por primera vez y cuándo se envía al servidor, lo que le permite realizar la programación en consecuencia.

Estructura (structure).- Tipo de valor definido por el usuario, al igual que una clase, las estructuras pueden contener constructores, constantes, campos, métodos, propiedades, indizadores, operadores y tipos anidados. Sin embargo, a diferencia de las clases, las estructuras no admiten la herencia.

Evaluador de comparaciones (comparison evaluator).- Filtro que compara el nombre de una función de dispositivo con un valor.

Evaluador delegado (delegated evaluator).- Filtro que utiliza un método personalizado para evaluar los datos especificados.

Expresión regular (regular expression).- Notación concisa y flexible para buscar y reemplazar modelos de texto. Esta notación incluye dos tipos de caracteres básicos: caracteres de texto literales (normales), que indican texto que debe existir en la cadena de destino, y metacaracteres, que indican el texto que puede variar en la cadena de destino. Puede utilizar expresiones regulares para analizar rápidamente grandes cantidades de texto con el fin de buscar modelos de caracteres específicos, para extraer, modificar, reemplazar o eliminar subcadenas de texto, o para agregar las cadenas extraídas a una colección con el fin de generar un informe.

F

Formularios Web Forms (Web Forms).- Marco de trabajo de página ASP.NET, compuesto por páginas Web programables (denominadas páginas de formularios Web Forms) que contienen controles de servidor reutilizables.

Framework.- Plataforma para crear, implementar y ejecutar aplicaciones y servicios Web XML. Proporciona un entorno de múltiples lenguajes basado en estándares y muy productivo para integrar las inversiones existentes con aplicaciones y servicios de la próxima generación, así como la agilidad necesaria para resolver los desafíos que suponen la implementación y el funcionamiento de las aplicaciones para Internet. .NET Framework consta de tres partes principales: Common Language Runtime, un conjunto jerárquico de bibliotecas de clases unificadas y una versión dividida en componentes de ASP denominada ASP.NET.

Función (role)- Conjunto de principales con nombre que tienen los mismos privilegios con respecto a la seguridad.

G

Gifs animados.- Es una secuencia de imágenes fijas, en formato gif, que vistas una detrás de otra producen una animación.

Globalización (globalization).- Proceso de diseñar y programar un producto de software de manera que funcione en varias configuraciones regionales. La globalización implica la identificación de las configuraciones regionales que se deben admitir, el diseño de funciones compatibles con dichas configuraciones regionales y la escritura de código que funcione igual de bien en cualquiera de las configuraciones regionales admitidas.

H

HTTP.- HyperText Transfer Protocol. Protocolo de enlace a páginas WEB.

HTML.- HyperText Mark-up Language. Lenguaje hipertexto utilizado para visualizar estas páginas.

Hoja de estilos externa (external style sheet): Hoja de estilos definida en un control de usuario de un archivo .ascx.

Hoja de estilos interna (internal style sheet).- Hoja de estilos incluida en una página de formularios Web Forms móviles de ASP.NET.

I

IIS.- Internet Information Service (Server).- Servidor Web de Microsoft Windows.

Interfaz (interface).- Tipo de referencia que define un contrato. Otros tipos implementan una interfaz para garantizar que admiten ciertas operaciones. La interfaz especifica los miembros que las clases u otras interfaces que los implementan deben suministrar. Al igual que las clases, las interfaces pueden contener como miembros métodos, propiedades, indizadores y eventos.

J

JAVA.- Lenguaje de Programación popularizado en Internet. Originalmente creado por *Sun Microsystems*.

JIT.- Acrónimo de "Just-In-Time", una expresión que describe una acción que sólo se realiza cuando es necesario, como la compilación Just-In-Time o la activación de objetos Just-In-Time.

L

Lenguaje de descripción de servicios Web (WSDL, Web Services Description Language).- Lenguaje de contrato basado en XML para describir los servicios de red ofrecidos por un servidor. Vea también: Lenguaje de marcado extensible (XML).

Lenguaje de marcado extensible (XML, Extensible Markup Language).- Subconjunto del Lenguaje de marcado generalizado estándar (SGML) optimizado para su uso a través del Web. XML proporciona un método uniforme para describir e intercambiar datos estructurados que es independiente de las aplicaciones o los proveedores.

Lenguaje intermedio de Microsoft (MSIL, Microsoft intermediate language).- Lenguaje utilizado como salida de una serie de compiladores y como entrada para un compilador Just-In-Time (JIT). Common Language Runtime incluye un compilador Just-In-Time para convertir MSIL a código nativo.

Localización (localization).- Proceso de personalizar o traducir los datos y los recursos necesarios para una región o un idioma determinados.

M

Módulo (module).- Unidad cargable que puede contener declaraciones e implementaciones de tipo. El módulo contiene información suficiente para que Common Language Runtime pueda encontrar todos los bits de implementación al cargar el módulo. El formato de los módulos es una extensión del formato de archivo

ejecutable portable (PE) de Windows. Cuando se implementa, un módulo está contenido siempre en un ensamblado.

Modo a prueba de errores (safe mode).- Directiva de versiones específica que solicita la ejecución de un ensamblado determinado con la versión exacta de sus dependencias con la que se compiló.

Metadatos (metadata).- Información que describe todos los elementos administrados por Common Language Runtime: un ensamblado, el archivo cargable, el tipo, el método, etc. Esto puede incluir información necesaria para la depuración y la recolección de elementos no utilizados, así como atributos de seguridad, cálculo de referencias de datos, definiciones extendidas de clases y miembros, enlace de versión y otra información requerida por el motor de tiempo de ejecución.

N

.NET Compact Framework.- Entorno independiente del hardware para ejecutar programas en dispositivos informáticos con recursos insuficientes. Hereda la arquitectura completa de .NET Framework de Common Language Runtime, es compatible con un subconjunto de la biblioteca de clases de .NET Framework y contiene clases diseñadas exclusivamente para .NET Compact Framework. Entre los dispositivos que admite se encuentran asistentes de datos personales (PDA) (como Pocket PC), teléfonos móviles, dispositivos de conexión para televisión digital, dispositivos informáticos de la industria del automóvil y dispositivos incrustados de diseño personalizado generados con el sistema operativo Microsoft Windows CE.NET.

.NET Framework.- Plataforma para crear, implementar y ejecutar aplicaciones y servicios Web XML. Proporciona un entorno de múltiples lenguajes basado en estándares y muy productivo para integrar las inversiones existentes con aplicaciones y servicios de la próxima generación, así como la agilidad necesaria para resolver los desafíos que suponen la implementación y el funcionamiento de las aplicaciones para Internet. .NET Framework consta de tres partes principales: Common Language Runtime, un conjunto jerárquico de bibliotecas de clases unificadas y una versión dividida en componentes de ASP denominada ASP.NET.

O

ODBC.- Controlador de Bases de Dato Tiene asociado un API que permite el manejo de bases de datos de forma independiente a la misma, ya que utiliza controladores específicos para cada base de datos.

Objeto de permiso (permission object).- Instancia de una clase de permiso que representa derechos de acceso a los recursos o la identidad. Un objeto de permiso puede utilizarse para especificar una solicitud, una demanda o la concesión de un permiso.

P

Página web: Es el formato estándar de la Red (aunque también hay otros), y son documentos escritos en HTML (y otros lenguajes) que, al ser transcritos por el navegador, se transforman en las pantallas que vemos en Internet.

Password: Es la palabra inglesa que se traduce en español como contraseña o clave.

Página ASP.NET (ASP.NET page).- Componente de una aplicación ASP.NET.

Paginación (pagination).-Mecanismo que separa de forma automática el contenido de los formularios Web Forms móviles de ASP.NET en grupos más pequeños de páginas procesadas destinadas a encajar con un dispositivo específico. También procesa elementos de la interfaz de usuario que se pueden utilizar para buscar otras páginas.

Permisos concedidos (granted permissions).- Permisos que se concederá al código, determinados por la directiva de seguridad, y que le permiten el acceso a recursos y le brindan su identidad. Los permisos concedidos están determinados tanto por los permisos solicitados como por lo que permite la configuración de la directiva de seguridad.

Permisos solicitados (requested permissions).- Permisos especificados opcionalmente en un ensamblado que representan los permisos mínimos necesarios, deseados opcionalmente, y que siempre se rechazan para todo el código del ensamblado. Si no hay ninguna solicitud, se concede al código el máximo que permita la directiva de seguridad.

Plantilla (template).- En ASP.NET: fragmento de página declarativo que se utiliza para proporcionar un control de servidor ASP.NET con plantilla a una interfaz visual. Una plantilla contiene elementos de presentación entre los que se incluyen texto literal, HTML y expresiones de enlace a datos, así como elementos de sintaxis declarativos que representan controles de servidor ASP.NET. Una plantilla puede persistir como un archivo de texto con una extensión .ascx.

Principal.- En la seguridad de .NET Framework, representa la identidad y la función de un usuario, y actúa en su nombre.

Poblema de seguridad (security hole).- Punto de entrada no protegido de forma no intencionada a un equipo, componente, aplicación u otro recurso en línea seguro.

Propiedad (property).- Miembro de clase que es como un campo público, pero que incluye funciones como control de versiones, encapsulación y la posibilidad de ejecutar otra lógica adicional mediante los métodos de descriptor de acceso **get** y **set**.

Proveedor de datos de .NET Framework (.NET Framework data provider).- Componente de ADO.NET que proporciona acceso a datos desde un origen de datos relacional. Un proveedor de datos de .NET Framework contiene clases para conectarse a un origen de datos, ejecutar comandos en el origen de datos y devolver resultados de consultas desde el origen de datos, e incluye la capacidad de ejecutar comandos en transacciones. Un proveedor de datos de .NET Framework también contiene clases para llenar un DataSet con resultados de un origen de datos y propagar los cambios de un DataSet en el origen de datos.

R

RAM.- (Random Access Memory) Memoria de acceso aleatorio y de tipo volátil o temporal. Es la memoria de trabajo de una PC.

Recurso (resource).- Cualquier dato no ejecutable que se implemente lógicamente con una aplicación. Los recursos pueden mostrarse en una aplicación como mensajes de error o como parte de la interfaz de usuario. Los recursos pueden contener datos con varios formatos, como objetos almacenados, cadenas e imágenes.

S

Scripts.- Es como se denomina al código que va entre dos etiquetas para indicar que pertenece a un lenguaje distinto a HTML. En una página web podemos escribir códigos de diversos lenguajes como JavaScript, Java, CGI, etc., no obstante para que el navegador sepa interpretarlo, irá siempre entre scripts o etiquetas que indiquen a qué tipo de lenguaje pertenece.

Server:.- Servidor.- Se refiere a un ordenador de gran potencia conectado a una red informática, que se encarga de recibir, procesar y devolver la respuesta a las peticiones que el usuario realiza desde su ordenador.

Seguridad de acceso a código (code access security).- Mecanismo proporcionado por Common Language Runtime en el que la directiva de seguridad concede permisos al código administrado y se exigen estos permisos, limitando las operaciones que el código puede realizar. Para impedir que las rutas de acceso a código no previstas presenten una vulnerabilidad de la seguridad, se debe conceder a todos los llamadores de la pila de llamadas los permisos necesarios (posiblemente sujetos a su reemplazo por una aserción o una denegación).

Servicios Web XML (XML Web services).- Unidades de lógica de aplicaciones que proporcionan datos y servicios a otras aplicaciones. Las aplicaciones obtienen acceso a los servicios Web XML mediante protocolos Web estándar y formatos de datos como HTTP, XML y SOAP, con independencia de cómo se implementa cada servicio Web XML. Los servicios Web XML combinan los mejores aspectos del desarrollo basado en componentes y el Web, por lo que son una base fundamental del modelo de programación de Microsoft .NET.

SOAP.- Protocolo simple basado en XML para intercambiar información estructurada y de tipos en el Web. El protocolo no contiene semántica de aplicación ni de transporte, por lo que resulta muy modular y extensible.

T

Tabla.- Es un conjunto de datos sobre un tema específico, como productos o proveedores. Si se usa una tabla independiente para cada tema evita la duplicación de datos, lo que hará que vuestra base de datos sea más eficiente y reduzca los errores de entrada de datos.

U

URL.- Localizador Uniforme de Recursos, es decir, la dirección *web*.

V

Viaje de ida y vuelta (round trip).- Secuencia de fases de Web Forms que comienza cuando una acción de usuario requiere procesamiento en el servidor. Cuando esto ocurre, se debe enviar al servidor un formulario, procesarlo y devolverlo al explorador.

W

World Wide Web.- Es el nombre que damos al conjunto de documentos que podemos encontrar en Internet. Dichos documentos, que habitualmente denominamos web, son multimedia, es decir, combinan textos, imágenes y sonidos.

Web Forms.- Formularios Web.- Biblioteca de clientes para Web.

Windows Forms.- Formularios Windows.- Biblioteca de clientes de Windows.

Web Page.- Página Web.- Página de un sitio que está navegando en la red.

WAP.- Protocolo estándar para proporcionar comunicaciones por Internet y servicios de telefonía avanzados para teléfonos, localizadores (pager), asistentes digitales personales (PDA) y otras terminales inalámbricas.

WML.- Lenguaje de marcado basado en XML que se utiliza para especificar el contenido y la interfaz de usuario de dispositivos de banda estrecha, incluidos los teléfonos celulares y los localizadores (pager). WML es parte de WAP.

X

XML.- Extensible Markup Language (XML) es un sencillo formato de texto flexible que puede utilizarse como base para crear nuevos lenguajes de marcado que pueden emplearse en la publicación de documentos y el intercambio de datos. XML se basa en una serie de recomendaciones publicadas por los grupos de trabajo del World Wide

Web Consortium (W3C) y, por tanto, es apto (aunque no se limita) para utilizarse en aplicaciones basadas en Web.

En Microsoft, los ingenieros contribuyen a los esfuerzos de los grupos de trabajo del W3C que definen y establecen normas para XML y otros protocolos Web. En la medida de lo posible, se intenta conseguir el completo cumplimiento de las normas XML una vez que el W3C redacta la versión definitiva y la publica como recomendación.

Entre las normas XML para las que Microsoft ofrece actualmente apoyo a los programadores se encuentran:

- El lenguaje de definición de esquema XML (XSD), una norma W3C actual para utilizar XML con el fin de crear esquemas XML. Los esquemas XML se pueden utilizar para validar otros documentos XML.
- Extensible Stylesheet Language Transformations (XSLT) 1.0, una norma W3C de lenguaje de hojas de estilo XML. XSLT está recomendada para transformar documentos XML.
- XML Path Language (XPath) 1.0, una norma XML de W3C que utiliza XSLT y otros lenguajes de programación XML para consultar y filtrar datos almacenados en documentos XML.

Sintaxis basada en .NET Framework versión 1.1.4322.

5.5. BIBLIOGRAFÍA.

5.5.1. Bibliografía Consultada

CHARTE O. Francisco, Programación con Visual C# .NET, Madrid, 2002.

Documentación de .NET Framework versión 1.1.4322.

HERNÁNDEZ, Roberto, Metodología de la Investigación, México, 2001.

LEIVA ZEA, Francisco, Nociones de Metodología de Investigación Científica, Cuarta Edición, Quito – 1996.

TAMAYO Y TAMAYO, Mario, El Proceso de la Investigación Científica, Tercera Edición, México 1997.

ULLOA, Francisco, Investigación 2000, Latacunga, 2004.

<http://www.microsoft.com/latam/msdn/comunidad/mtj.net/2002/07/art01/default.as>

5.5.2. Bibliografía Citada

CHARTE O. Francisco, Programación con Visual C# .NET, Madrid, 2002, pp. 540, 545, 546.

KUO, Benjamín C., Sistemas Automáticos de Control, México, 1996. pp. 16.

KROENKE, David M., Procesamiento de Base de Datos: Fundamentos, Diseño e Implementación, México, 1996. pp. 3.

LOOMIS, Mary E. S., Estructura de Datos y Organización de Archivos, México, 1991. xxi.

MUNICH, Lourdes, Métodos y Técnicas de Investigación, México, 1990. pp 13.

NAMAKFOROOSH, Mohammad Naghi, Metodología de la Investigación, México 1989, pp.27.

STONER, James, Administración, México, 1996. pp. 7.

VARELA, Villegas Rodrigo, Evaluación económica de Proyectos de Inversión, 1997. pp. 4.

ZINKMUND, William G., Investigación de Mercados, México, 1998. pp. 7.

5.6. ANEXOS.