

UNIVERSIDAD TÉCNICA DE COTOPAXI
CARRERA DE CIENCIAS DE LA INGENIERÍA Y APLICADAS
ESPECIALIDAD INGENIERÍA EN INFORMÁTICA Y CIENCIAS
COMPUTACIONALES.

Sr. Ing.

Guido Yauli
DIRECTOR DE LA CARRERA
Presente

De mi consideración.

Por medio de la presente le hago extensible el informe de la Tesis de los Señores:

AUTORES:

SEGOVIA ACOSTA IVAN MARCELO
SEGOVIA ZAMBRANO GUIDO ORLANDO

DIRECTOR:

DR. MARCELO BAUTISTA

TEMA:

**DESARROLLO DE UN SOFTWARE CLIENTE SERVIDOR
PARA EL CONTROL Y GESTIÓN DE LA INFORMACIÓN
ACADÉMICA DE LA CARRERA DE EDUCACIÓN
SEMIPRESENCIAL, ESPECIALIDAD “EDUCACIÓN BÁSICA”
PARA LA UNIVERSIDAD TÉCNICA DE COTOPAXI EN EL
CENTRO ASOCIADO LATACUNGA.**

Revisado el documento de la Tesis de los señores antes mencionados pude encontrar algunas observaciones las cuales deben ser corregidas antes del día de la defensa de Tesis.

- El diseño de aplicación debe guardar un mismo esquema para lo que es atributos de las aplicaciones.
- Existe redundancia en los conceptos de las herramientas que se van a utilizar se sugiere que se le deje únicamente como MARCO TEÓRICO.
- Dentro de la aplicación faltan muchos Reportes los cuales deberían ir en beneficio de las personas que van a utilizar, lo ideal pudiera ser cuando menos un reporte por estudiante, uno por curso, uno por especialidad.
- Cuando se manipula información únicamente se tiene un usuario que viene siendo el administrador del Sistema, lo cual podría ocasionar ciertos daños

por mala manipulación, lo sugerido sería crear un usuario donde se pueda darle únicamente permisos de usuario, para ingreso de notas únicamente.

▪ **INDICE GENERAL**

- **PORTADA**
- **DERECHOS DE AUTORIA**
- **CERTIFICACIÓN DEL DIRECTOR DE TESIS**
- **AGRADECIMIENTO**
- **DEDICATORIA**
- **INDICE GENERAL**
- **RESUMEN**
- **SUMARY**
- **CERTIFICADO DEL SUMARY**
- **INTRODUCCIÓN**
- **CAPITULO I**
- **LENGUAJES DE PROGRAMACIÓN Y SISTEMAS OPERATIVOS 1**
- **Introducción a Visual Studio .NET. ¿QUÉ ES .NET?..... 1**
- **Visual Studio. NET 1**
- **1.1.1.1 Modelo de programación consistente 2**
- **1.1.1.1.3 Eliminación del “infierno de las DLLs” 3**
- **1.1.1.2 Modelo de programación sencillo 3**
- **1.1.1.4 Integración de lenguajes..... 3**
- **1.1.1.5 Framework 1.1 .Net..... 4**
- **1.1.1.6 Seguridad avanzada..... 5**
- **1.1.1.7 Requisitos de Hardware 5**
- **1.1.2 Visual Basic.Net..... 8**

▪ 1.1.2.1 Beneficios de Visual Basic .Net.....	9
▪ 1.1.3 Visual C++. Net.	9
▪ 1.1.3.1 Beneficios.....	9
▪ 1.1.4 Visual C#.Net.....	10
▪ 1.1.4.1 lenguaje nativo de .NET.....	11
▪ 1.1.4.2 Características de C#.....	11
▪ 1.1.4.3 Sencillez:	12
▪ 1.1.4.4 Modernidad	12
▪ 1.1.4.5 Orientación a objetos.....	13
▪ 1.1.4.6 Orientación a componentes	13
▪ 1.1.4.7 Instrucciones seguras	13
▪ 1.1.4.8 Eficiente	14
▪ 1.1.4.9 Orientación a objetos:	14
▪ 1.1.4.10 Orientación a componentes:.....	16
▪ 1.1.4.11 Introducción a las páginas de formularios Web Forms.....	16
▪ 1.1.4.11.1 Componentes de los formularios Web Forms.....	17
▪ 1.1.4.11.2 Ventajas que aportan las páginas de formularios Web Forms .	19
▪ 1.1.4.11.3 Derivar de la clase Page	21
▪ 1.1.4.11.4 Elementos de las aplicaciones Web ASP.NET	23
▪ 1.1.4.12 ADO.Net	25
▪ 1.1.4.12.1 ADO.NET no depende de conexiones continuamente activas.	25
▪ 1.1.4.12.2 Las interacciones con la base de datos se realizan mediante comandos de datos	26

▪ 1.1.4.12.3 Los conjuntos de datos son independientes de los orígenes de datos	27
▪ 1.1.4.12.4 Componentes de ADO.NET.....	27
▪ 1.2 SISTEMAS OPERATIVOS PARA SERVIDORES	28
▪ 1.2.1 Sistema Operativo Windows Xp Profesional.....	28
▪ 1.2.1.1 Introducción	28
▪ 1.2.1.2 Características de Windows XP Profesional.....	29
▪ 1.3 BASE DE DATOS.....	32
▪ 1.3.1 Qué es una Base de Datos	32
▪ 1.3.2 Objetos de Base de Datos Relacionales.	33
▪ 1.3.3 Concepto de SQL Server.....	35
▪ 1.3.3.1 Introducción a SQL Server y a las Bases de Datos Relacionales.	35
▪ 1.3.3.2 Características de SQL Server	35
▪ 1.3.3.3 SQL Server y el Modelo Cliente / Servidor.	36
▪ 1.3.3.3.1 Ventajas de los clientes.	37
▪ 1.3.3.3.2 VENTAJAS DEL SERVIDOR.....	37
▪ 1.3.4. RDBMS.....	38
▪ 1.3.4.1 ¿Que es un RDBMS?	38
▪ CAPITULO II	
▪ TRATAMIENTO DE LA INFORMACIÓN DE LA ACTIVIDAD ACADÉMICA DE LA CARRERA DE EDUCACIÓN SEMIPRESENCIAL EN EL CENTRO ASOCIADO LATACUNGA	39
▪ 2.1 LA CARRERA DE EDUCACIÓN SEMIPRESENCIAL EN LA UTC.	39

▪ 2.1.1 Reseña Histórica de la Carrera	39
▪ 2.1.2 MISIÓN	39
▪ 2.1.5 POLÍTICAS	40
▪ 2.1.6 ESTRATEGIAS	41
▪ 2.1.7 METODOLOGÍA	42
▪ 2.1.8 RECURSOS	43
▪ 2.1.8.1 HUMANOS	43
▪ 2.1.8.2 INFRAESTRUCTURA	43
▪ 2.1.9 REGLAMENTO PARA EL FUNCIONAMIENTO DEL AREA DE ESTUDIO A DISTANCIA	43
▪ 2.1.9.1 CAPITULO I (Artículos de la ley de la Carrera)	45
▪ 2.1.9.2 DE LOS DOCENTES	49
▪ 2.1.9.3 DE LOS ESTUDIANTES	50
▪ 2.1.9.4 DE LA EVALUACION Y ACREDITACION	51
▪ 2.1.9.5 DE LA GRADUACION Y CERTIFICACION	52
▪ 2.1.9.6 DE LOS COSTOS Y FINANCIAMIENTO	52
▪ 2.2 EL MANEJO DE LA INFORMACIÓN EN LA CARRERA DE EDUCACIÓN SEMIPRESENCIAL, EN LA AC	54
▪ 2.2.1 OBJETIVOS DE LA CARRERA	54
▪ 2.2.2 PERFIL PROFESIONAL	55
▪ 2.2.3 REQUISITOS: (PRIMER CICLO)	56
▪ 2.3 CRITERIOS DE LOS DOCENTES	57
▪ 2.4 CRITERIOS DE LOS ESTUDIANTES	70
▪ CAPÍTULO III	

<ul style="list-style-type: none"> ▪ DISEÑO DE SOFTWARE PARA EL MANEJO DE BASE DE DATOS CON INFORMACIÓN DE LA ACTIVIDAD ACADÉMICA DE LA CARRERA DE EDUCACIÓN SEMIPRESENCIAL EN EL CENTRO ASOCIADO LATACUNGA DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI 	83
<ul style="list-style-type: none"> ▪ 3.1.1 Modelo de Fases..... 	83
<ul style="list-style-type: none"> ▪ 3.1.1.1 Fase de Planificación y Especificación de Requisito..... 	83
<ul style="list-style-type: none"> ▪ 3.1.1.1.1 Planificación..... 	84
<ul style="list-style-type: none"> ▪ 3.1.1.1.1.1 Propósito 	84
<ul style="list-style-type: none"> ▪ 3.1.1.1.1.2 Ámbito del Sistema..... 	84
<ul style="list-style-type: none"> ▪ 3.1.1.1.1.3 Alcance del Sistema 	84
<ul style="list-style-type: none"> ▪ 3.1.1.1.1.4 Definición de Usuarios..... 	86
<ul style="list-style-type: none"> ▪ 3.1.1.1.2 Descripción General..... 	87
<ul style="list-style-type: none"> ▪ 3.1.1.1.2.1 Funciones del Sistema..... 	87
<ul style="list-style-type: none"> ▪ 3.1.1.1.3 Requisitos Específicos 	88
<ul style="list-style-type: none"> ▪ 3.1.1.1.3.1 Requisitos Funcionales 	88
<ul style="list-style-type: none"> ▪ 3.1.1.1.3.1.1 Gestión de Materias 	88
<ul style="list-style-type: none"> ▪ 3.1.1.1.3.1.2 Gestión de Docentes..... 	88
<ul style="list-style-type: none"> ▪ 3.1.1.1.3.1.3 Gestión de Alumnos..... 	89
<ul style="list-style-type: none"> ▪ 3.1.1.1.3.1.4 Gestión de Notas 	89
<ul style="list-style-type: none"> ▪ 3.1.1.1.3.1.5 Gestión de Faltas 	90
<ul style="list-style-type: none"> ▪ 3.1.1.1.3.1.6 Gestión de Historial 	90
<ul style="list-style-type: none"> ▪ 3.1.1.1.3.1.7 Gestión de Cierre del Sistema..... 	90
<ul style="list-style-type: none"> ▪ 3.1.1.1.3.1.8 Gestión de Egresados 	91

▪ 3.1.1.1.3.2 Requisitos de Rendimiento	91
▪ 3.1.1.1.3.3 Requisitos Tecnológicos	91
▪ 3.1.1.1.3.3.1 Requisitos de Hardware	91
▪ 3.1.1.1.3.3.2 Requisitos de Software.....	92
▪ 3.1.1.1.3.4 Requisitos de Desarrollo	92
▪ 3.1.1.1.3.5 Requisitos de Comunicación.....	92
▪ 3.1.1.1.3.6.1 Seguridad	93
▪ 3.1.1.1.3.6.2 Tipos de Usuarios.....	93
▪ 3.1.1.1.3.6.2.1 Administrador del Sistema.....	93
▪ 3.1.1.1.3.6.2.3 Docentes y Alumnos	94
▪ 3.1.1.2 Diseño	94
▪ 3.1.1.2.2 Diagrama de Secuencias	96
▪ 3.1.1.2.3 Diagrama de Clases.....	98
▪ 3.1.1.2.4 Diseño Lógico	99
▪ 3.1.1.2.5 Diseño Físico.....	101
▪ 3.1.1.2.6 Diccionario de datos.....	102
▪ 3.1.1.3 Prueba del Sistema	106
▪ 3.1.1.3.1 Prueba de Datos	107
▪ 3.1.1.3.2 Prueba de Validaciones	108
▪ 3.1.1.3.3 Prueba de Seguridades	109
▪ 3.1.1.4 Implantación del Sistema	111
▪ CONCLUSIONES	
▪ RECOMENDACIONES	
▪ GLOSARIO	

- **BIBLIOGRAFIA**

- **ANEXOS**

RESUMEN

En la actualidad, las nuevas tecnologías de la información y comunicación se han convertido en herramientas imprescindibles para mejorar cualitativamente la gestión de la información. En tal sentido la presente investigación “DESARROLLO DE UN SOFTWARE CLIENTE SERVIDOR PARA EL CONTROL Y GESTIÓN DE LA INFORMACIÓN ACADÉMICA DE LA CARRERA DE EDUCACIÓN SEMIPRESENCIAL, ESPECIALIDAD “EDUCACIÓN BÁSICA” PARA LA UNIVERSIDAD TÉCNICA DE COTOPAXI EN EL CENTRO ASOCIADO LATACUNGA” constituye un avance para la institución en el aprovechamiento y utilización de procesos modernos en la administración universitaria.

Por esta razón se plantea desarrollar e implementar un software para el manejo de información de la Carrera Semipresencial en la Especialidad “Educación Básica” del Centro Asociado Latacunga, que permita automatizar los diferentes procesos académicos que se lleva a cabo en esta dependencia para estar a la vanguardia de los avances informáticos de la sociedad contemporánea.

Mediante la implementación de esta aplicación se puede determinar que los resultados obtenidos son medibles no solo en la cantidad de procesos que se desarrollan sino además en la calidad de la prestación de los servicios a los usuarios y estamentos universitarios; entre los cuales se menciona: la descarga de módulos correspondientes a las materias, consulta de notas individualizada vía Intranet, automatización general del proceso académico de la secretaría.

Esta investigación aporta al desarrollo institucional de la Universidad Técnica de Cotopaxi con el fin de incorporar nuevos avances de la informática para de esta manera favorecer el mejoramiento de la Carrera Semipresencial Especialidad Educación Básica.

Finalmente la contribución tanto de la investigación como de sus resultados constituyen una significación práctica mediante la implementación de este software; además genera las posibilidades a futuro del desarrollo de nuevos sistemas similares que ayuden en la utilización de las nuevas tecnologías de la información y comunicación aplicadas al proceso de enseñanza aprendizaje.

CAPÍTULO I

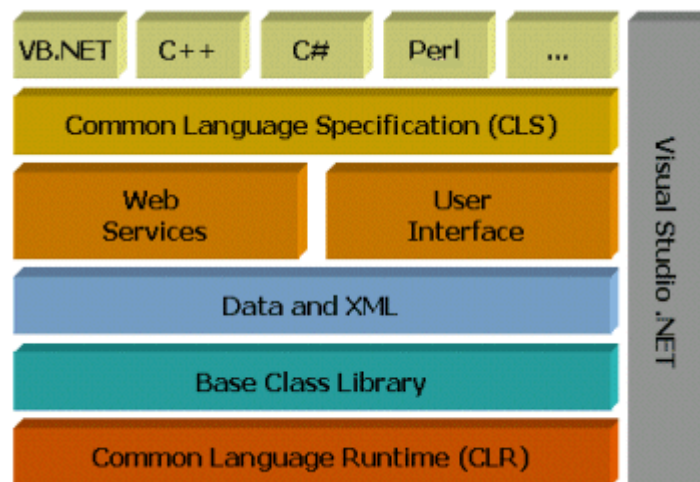
LENGUAJES DE PROGRAMACIÓN Y SISTEMAS OPERATIVOS

1.1 Visual Studio. NET

1.1.1 Introducción a Visual Studio .NET. ¿QUÉ ES .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 comentados 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 usa *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.

1.1.1.1 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)

1.1.1.2 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

1.1.1.3 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.

1.1.1.4 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 esta 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.

1.1.1.5 Framework 1.1 .Net

Windows .NET Framework 1.1 es el componente de Microsoft Windows® que permite crear y ejecutar la próxima generación de aplicaciones de software y servicios Web XML, componentes que facilitan la integración compartiendo datos e información a través de la red mediante protocolos estándar independientes de la plataforma, como por ejemplo, XML, SOAP y HTTP.

Windows .NET Framework ofrece lo siguiente:

- Un entorno enormemente productivo y basado en estándares para integrar el código existente con aplicaciones y servicios de próxima generación.
- La flexibilidad para afrontar los retos que suponen la implementación y la operación de aplicaciones en toda la empresa.

Windows .NET Framework se compone en dos elementos principales: Common Language Runtime y un conjunto unificado de bibliotecas de clases, tales como ASP.NET para aplicaciones Web y servicios Web XML, Windows Forms para aplicaciones cliente inteligentes y ADO.NET para el acceso a datos sin rigidez.

1.1.1.6 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 códigos 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.

1.1.1.7 Requisitos de Hardware

Para instalar las distintas ediciones de Visual Studio .NET, las ediciones Standard disponibles para los lenguajes de programación de Visual Studio .NET y sugerencias para mejorar el rendimiento de Visual Studio .NET¹.

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

Requisito	Enterprise Architect	Enterprise Developer	Professional	Academic
Procesador	PC con procesador de tipo Igual Pentium II, 450 MHz Se recomienda: tipo Pentium III, 600 MHz ¹	Igual	Igual	Igual

¹ UTC, Ayuda MSDN; Microsoft Visual Studio. Net; ms-help://MS.VSCC.2003/MS.MSDNQTR.2003FEB.3082/vsintro7/html/vxconATourOfVisualStudio.htm; año 2003

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 sistema y 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 <i>Se recomienda:</i> color de alta densidad, 16 bits	Igual	Igual	Igual

<i>Mouse</i> (ratón)	Microsoft	Mouse	o Igual	Igual	Igual
	dispositivo	señalador			
	compatible				

- Cuando 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.
- Windows NT 4.0 sólo admite la instalación de componentes remotos. Visual Basic, Visual C++, Visual C#, y Visual J# no se pueden instalar con Windows NT 4.0. Para obtener más información sobre los requisitos de hardware para otros productos incluidos en algunas ediciones, vea los archivos Léame correspondientes asociados.
- Windows XP Home no admite el desarrollo local de aplicaciones Web, solamente las versiones Professional y Server de Windows admiten esta función.
- Microsoft Windows 2000 Datacenter Server no es un sistema operativo compatible.
- El tipo de medios que se distribuyen con el producto determinan si se necesitará un CD-ROM o un DVD-ROM.

1.1.2 Visual Basic.Net

Visual basic.net es al nueva versión del conocido lenguaje de programación visual Basic Microsoft a querido reprogramar por completo el lenguaje de programación mas extendido del mundo y añadido en las características comunes a todo los leguajes de programación que forman parte de punto net.

La historia visual Basic se remota al año de 1991 hace mas de 10 años y desde entonces, este lenguaje de desarrollo de ha ido actualizando y mejorando para constituir un entorno de desarrollo actual que solucionan las demandas de los arquitectos y desarrolladores de hoy, con ASP.NET se ha querido ir bastante mas lejos y se ha querido olvidar del uso de VBScript.

1.1.2.1 Beneficios de Visual Basic .Net

Con la creación del entorno multiprogramación integrado .NET ha aparecido Visual basic.NET que entre otras características en la capacidad de ser un lenguaje POO(programación orientada a objetos) capacidades de crear servicios Web XML y otras características que permiten regenerar la imagen de un lenguaje de desarrollo completamente renovado.

1.1.3 Visual C++. Net.

Casi todas las clases de software del mundo, desde aplicaciones de consumo y juegos hasta aplicaciones corporativas de misión crítica, están construidas en Microsoft Visual C++. Visual C++ .NET es la más potente herramienta C++ para la creación de softwares de alta performance para Windows y Web.

1.1.3.1 Beneficios:

Visual C++ .NET brinda un Nuevo nivel de productividad a C++, sin sacrificar flexibilidad, performance, o control. Además de características como la Tecnología IntelliSense y Editar y Continuar con significativa velocidad en el proceso de desarrollo, Visual C++ .NET contiene un soporte mejorado para el desarrollo de aplicaciones Windows y Web².

Visual C++ .NET contiene un conjunto de nuevas y mejoradas características que permiten a los desarrolladores, crear y consumir XML Web Services y aplicaciones Web.

1.1.4 Visual C#.Net

Esta es la "vedette" de Visual Studio .NET, el nuevo lenguaje inspirado en C/C++, similar a Java.

Este lenguaje tiene una sintaxis basada en C/C++ con una estructura similar a Java pero con características especiales que lo hacen muy estructurado, sencillo y poderoso. Hágase la idea de que escribir en C# es como hacerlo en Visual Basic tradicional pero con resultados más poderosos, como lo esperaría de una aplicación C++.

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

² UTC, Ayuda MSDN; Microsoft Visual Studio. Net; ms-help://MS.VSCC.2003/MS.MSDNQTR.2003FEB.3082/vsintro7/html/vxoriVisualStudioNETSystemRequirements.htm; año 2003

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.

Aunque es posible escribir código para la plataforma

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.

1.1.4.1 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.

En resumen, 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. El hecho de ser relativamente reciente no implica que sea inmaduro, pues Microsoft ha escrito la mayor parte de la BCL usándolo, por lo que su compilador es el más depurado y optimizado de los incluidos en el *.NET Framework SDK*.

1.1.4.2 Características de C#

Con la idea de que los programadores más experimentados puedan obtener una

visión general del lenguaje, a continuación se recoge de manera resumida las principales características de C#. Algunas de las características aquí señaladas no son exactamente propias del lenguaje sino de la plataforma .NET en general³. Sin embargo, también se comentan aquí también en tanto que tienen repercusión directa en el lenguaje, aunque se indicará explícitamente cuáles son este tipo de características cada vez que se toquen:

1.1.4.3 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.

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 (::)

1.1.4.4 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

³ UTC, Internet; <http://www.programacion.com/tutorial/csharp/3>; Último ingreso: viernes 14 de noviembre 2003

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.

1.1.4.5 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.**

1.1.4.6 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)

1.1.4.7 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 aritmética⁴, 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.

1.1.4.8. Eficiente: 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 grandes.

1.1.4.9 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 presenta 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:

⁴ UTC, Internet; GONZÁLEZ, José Antonio; El lenguaje de programación C# ; año 2000

- 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.⁵ 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.

⁵ UTC, Internet; <http://microsoft.public.dotnet.languages.csharp>; Último ingreso: lunes 08 de diciembre 2003

1.1.4.10 Componentes de los formularios Web Forms

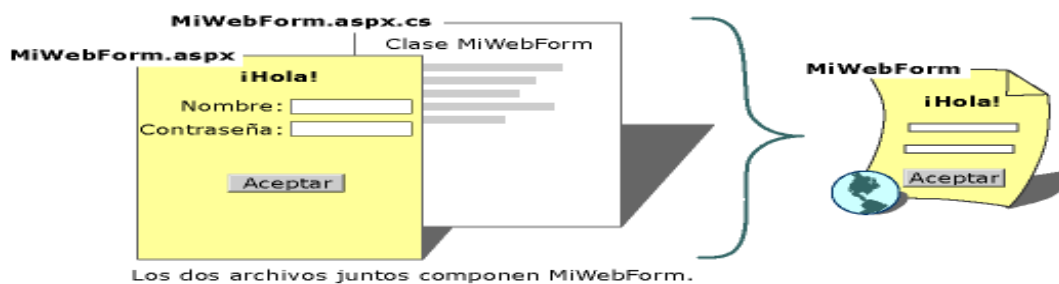
En las páginas de formularios Web Forms, la programación de la interfaz de usuario se divide en dos partes independientes: el componente visual y el lógico. Esta división entre la parte visible de un formulario y el código que se oculta detrás y que interactúa con él.

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 2. 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.

1.1.4.10.1 Ventajas que aportan las páginas de 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:

- **Implementar una interfaz de usuario Web enriquecida.** ⁶Una 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

⁶ UTC, Ayuda; MSDN; Microsoft Visual Studio. Net; ms-help://MS.VSCC.2003/MS.MSDNQTR.2003FEB.3082/vbcon/html/vbconIntroductionToWebForms.htm, año 2003

aplicaciones que deban ejecutarse en muchos exploradores y plataformas de dispositivos clientes distintos.

- **Separación entre cliente y 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.
- **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.
- **Complicaciones 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.
- **Complicaciones 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.

1.1.4.10.2 Derivar de la clase Page

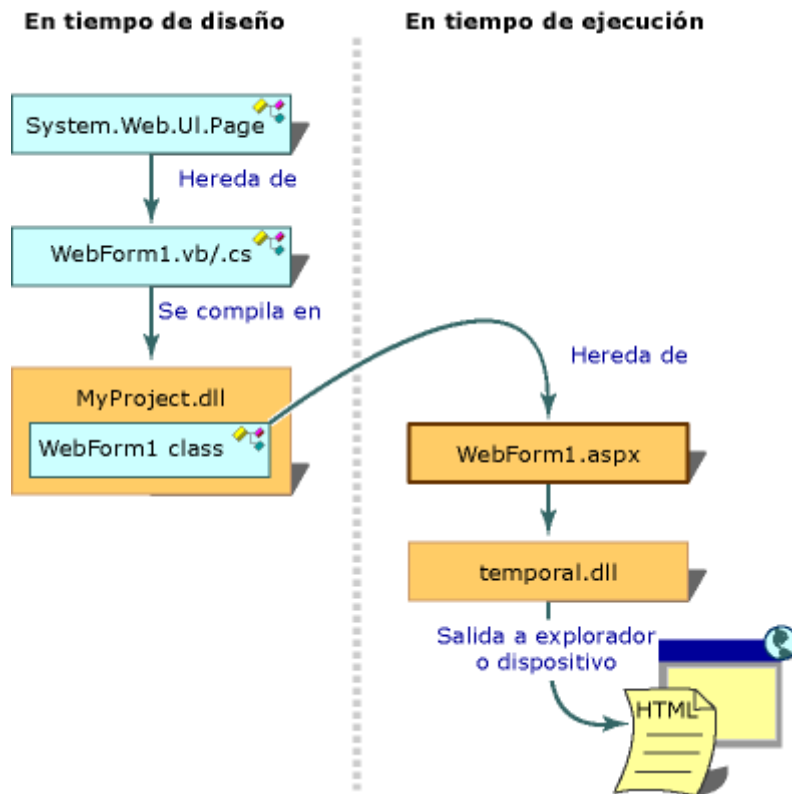
Cuando en Visual Studio crea los archivos de página y de clase de una página de formulario Web Forms, genera código que hereda de la clase **Page** básica. Por ejemplo, si se crea una nueva página de formularios Web Forms y se la denomina WebPage1, se derivará una nueva clase denominada WebPage1 de **System.Web.UI.Page**.

Dependiendo de si se ha elegido Visual Basic o 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 3 Estructura de la página de formularios Web Forms y la clase Page básica



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

⁷Debido 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"
```

⁷ UTC, Ayuda; MSDN; Microsoft Visual Studio. Net; ms-ms-help://MS.VSCC.2003/MS.MSDNQTR.2003FEB.3082/vbcon/html/vbconWebFormsCodeModel.htm, año 2003

Codebehind="WebForm1.aspx.cs" Inherits="WebApplication1.WebForm1"%>

1.1.4.10.3 Elementos de las aplicaciones Web ASP.NET

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:

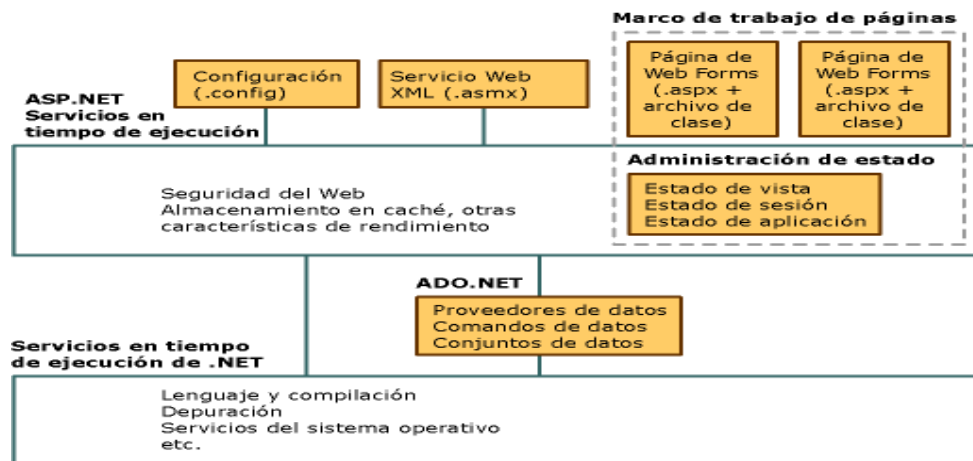
- **Funciones de administración de proyectos** Al crear una aplicación Web ASP.NET, necesitará hacer un seguimiento de los archivos necesarios, cuáles se debe compilar y cuáles necesitan implementarse.
- **Interfaz de usuario** Una aplicación suele presentar información a los usuarios; en una aplicación 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.
- **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, por ejemplo.
- **Datos** ⁸La 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.

⁸ UTC, Ayuda; MSDN; Microsoft Visual Studio. Net; ms-help://MS.VSCC.2003/MS.MSDNQTR.2003FEB.3082/vbcon/html/vbconIntroductionToWebApplicationsInVisualStudio.htm, año 2003

- **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.

En la figura 4 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 Framework.

Figura 4 Elementos de las aplicaciones Web ASP.NET



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

1.1.4.11 ADO.Net

1.1.4.11.1 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 muchas aplicaciones:

- Las conexiones abiertas de base de datos consumen valiosos recursos 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.
- Por la misma razón, las aplicaciones que necesitan una conexión de base de datos abierta resultan extremadamente 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, en particular, 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; cuando el servidor termina 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 permanecen inactivas, así que puede dar servicio a muchos más usuarios.

1.1.4.11.2 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.

1.1.4.11.3 Los conjuntos de datos son independientes de los orígenes de datos

⁹Aunque 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. Una

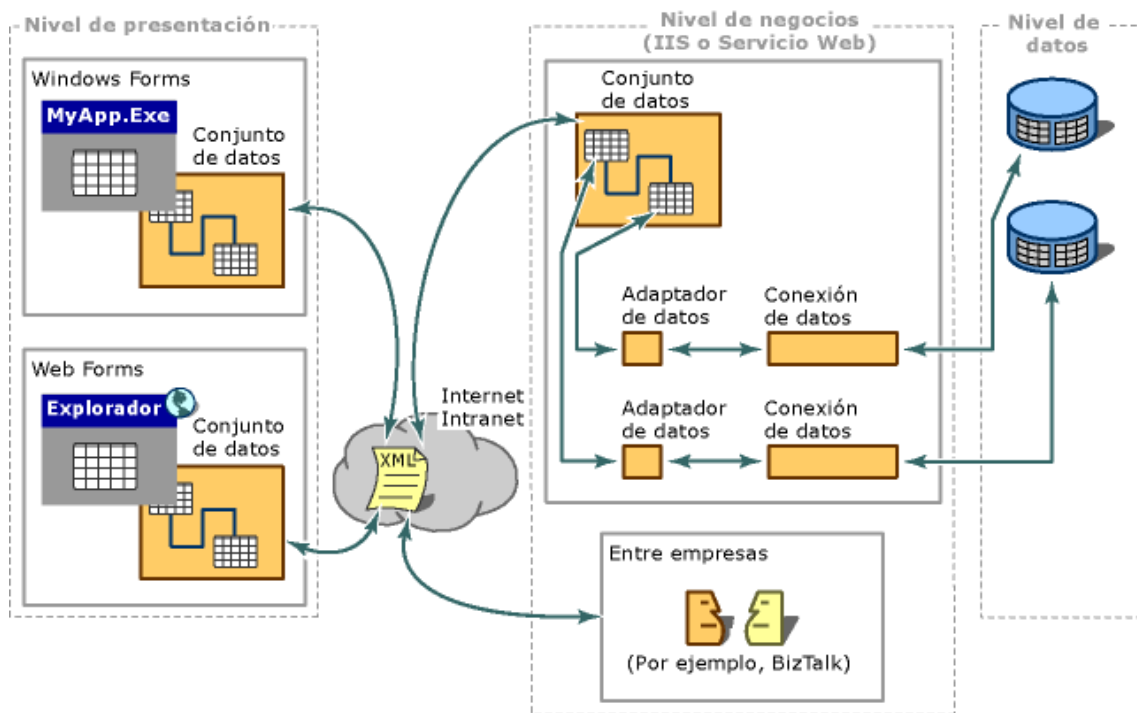
⁹ UTC, Biblioteca; CHARTE, O. Francisco; Programación con Visual C#. NET; Madrid, 2002

vez que los datos se encuentren en un conjunto de datos, podrá trabajar con ellos mediante un modelo de objetos coherente, independientemente de cuál sea su origen.

1.1.4.11.4 Componentes de ADO.NET

La Fig 5 muestra los componentes principales de una aplicación ADO.NET.

FIGURA 5 Componentes de datos 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, el sucesor 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.

Proporciona herramientas de productividad avanzadas que su organización necesita y le ofrece el poder de administrar, implementar y soportar su ambiente de computación de una manera más eficiente que nunca. Con Windows XP Profesional, Microsoft presenta un nuevo estándar para la computación eficiente y fiable. 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.

1.2.1.2 Características de Windows XP Profesional.

Las características se describen a continuación:

- ❖ **Desarrollado tomando como base el núcleo de Windows 2000.**

Windows XP Profesional tiene como base el 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.

- ❖ **Verificador optimizado de drivers de dispositivos.**

Desarrollado con base en el verificador de drivers de dispositivos de Windows 2000, la versión de Windows XP Profesional proporcionará pruebas más fuertes para drivers de dispositivos.

❖ **Escenarios de reinicio sustancialmente reducidos.**

Elimina la mayoría de los escenarios que obligaban a los usuarios finales a reiniciar en Windows NT 4.0 y Windows 9x. Así mismo, muchas de las instalaciones de software no requerirán de reinicio.

❖ **Protección mejorada del código.**

En Windows XP Profesional, las estructuras críticas de datos del kernel son de sólo lectura, de tal forma que los drivers y aplicaciones no los puedan afectar. Adicionalmente, todo el código del driver de dispositivos es de sólo lectura y está protegido.

❖ **Soporte paralelo DLL.**

Windows XP Profesional proporciona un mecanismo para que se instalen y ejecuten de manera paralela varias versiones de componentes individuales de Windows.

❖ **Protección de archivos de Windows.**

Windows XP Profesional protege los archivos del sistema para que no los sobrescriban las instalaciones de las aplicaciones. En el caso de que se sobrescriba un archivo, la Protección de archivos de Windows lo reemplazará con la versión correcta.

Rendimiento

❖ **Arquitectura de multitareas preventivas.**

Windows XP Profesional está diseñado para permitir que varias aplicaciones se ejecuten simultáneamente, al tiempo que asegura un excelente tiempo de respuesta y estabilidad del sistema.

❖ **Memoria escalable y soporte de procesadores.**

Soporta hasta 4 GB de memoria y hasta dos multiprocesadores simétricos..

Seguridad

❖ **Sistema encriptador de archivos (EFS) con soporte para multiusuarios.**

Encripta cada archivo con una clave generada aleatoriamente. Los procesos de encriptación y desencriptación son transparentes para el usuario. Con Windows XP Profesional, EFS ahora soporta la capacidad de que varios usuarios tengan acceso a un documento encriptado.

❖ **Seguridad IP (IPSec).**

Ayuda a proteger los datos que se transmiten a través de una red. IPSec es parte importante, debido a que proporciona seguridad para redes privadas virtuales (VPNs), que permiten a las organizaciones transmitir datos de manera segura por Internet.

Movilidad

❖ **Escritorio Remoto ("Remote Desktop").**

El Escritorio Remoto permite a un usuario crear una sesión virtual en su ordenador de escritorio utilizando el Protocolo de escritorio remoto (RDP) de Microsoft.

❖ **Administrador de credenciales.**

Un administrador de credenciales es un depósito seguro para información de claves de acceso. Esta función le 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.

1.3 BASE DE DATOS

1.3.1 Qué es una Base de Datos

Son aquellas en las que los datos están organizados en tablas las mismas que están organizadas agrupando datos acerca del mismo tema y contienen columnas y filas de información. Las tablas se relacionan de nuevo entre sí por medio del motor de base de datos cuando se solicita.

En SQL SERVER una base de datos no necesariamente está atada a un archivo, es más bien un concepto lógico con base en un conjunto de objetos relacionados. Por ejemplo en SQL Server, una base de datos contiene no solo los datos, sino además la estructura de la base de datos, todos los índices, la seguridad de la base de datos y tal vez otros objetos como vistas o procedimientos almacenados, relacionado con esa base de datos en particular.

1.3.2 Objetos de Base de Datos Relacionales.

Los siguientes son algunos de los objetos más comunes:

- **Tablas.-** Son los objetos que tienen los tipos de datos y los datos en sí.
- **Columnas.-** Son las partes de la tablas que contienen los datos, a las columnas debe asignarles un tipo de datos y un nombre único.
- **Tipos de datos.-** Hay varios tipos de datos como: de carácter, numérico o de fecha. A una columna en una tabla se le asigna un solo tipo de datos.
- **Procedimientos Almacenados.-** Son similares a las macros que se utiliza en productos como Excel o Word de Microsoft en que se puede escribir código Transact-SQL y almacenamiento bajo un nombre. Al ejecutar el procedimiento almacenado se está ejecutando el código Transact-SQL contenido en él.

¹⁰Un uso sería tomar el código Transact-SQL que ejecuta un informe semanal guardarlo como un procedimiento almacenado y a partir de ahí ejecutar solamente el procedimiento almacenado para generar el informe, también pueden emplearse procedimientos almacenados como mecanismos de seguridad.

- **Disparadores.-** Son procedimientos almacenados que se activan cuando se agregan modifican o eliminan datos de la base de datos, se emplean para asegurar que se respeten en las bases de datos las reglas empresariales u otro tipo de regla de integridad de datos.
- **Reglas.-** Se asignan a las columnas de modo que los datos que se registren deban apearse a los estándares que se establezca.

¹⁰ UTC, Internet; <http://www.microsoft.com/catalog/display.asp?subid=45&site=10798&x=37&y=1>; Último acceso: jueves 18 de diciembre 2003

- Claves principales.- Aunque no son objetos en sí, las claves son esenciales para las bases de datos relacionales, estas claves hacen que se cumplan su carácter único entre las filas proporcionando una forma de identificar de manera única cada elemento que se desea almacenar.
- Claves externas.- Una vez más sin ser realmente objetos estas claves son columnas que hacen referencia a claves principales o restricciones únicas de otras tablas. SQL Server usa las claves principales y externas para relacionar datos juntos a partir de tablas separadas al efectuar consultas.
- Restricciones.- Son mecanismos de obligatoriedad-integridad de datos implementada por el sistema con base en el servidor.
- Valores Predeterminados.- Se pueden asignar valores predeterminados a los campos de modo que sino se registran datos durante una operación Insert en una columna específica, se utiliza en dichos valores.
- Vistas.- Son consultas almacenadas en las bases de datos que pueden hacer referencias a una o varias tablas, se puede crearlas y guardarlas a fin de utilizarlas con facilidad en el futuro. Las vistas excluyen ciertas columnas de una tabla o bien vinculan varias tablas, también se puede emplearlas como mecanismos de seguridad para que ciertos usuarios puedan consultarlos y otros no.
- Índices.- Puede ayudarle a organizar los datos a efecto de que las consultas se ejecuten con mayor rapidez.

1.3.3 Concepto de SQL Server

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

Se inicia sobre antecedentes de SQL SERVER y Windows(95,98 y NT). Se verá las bases de datos y lo que conforma un ambiente cliente / servidor, después se continuará con las bases de datos y sus contenidos por último se revisará el diseño de base de datos.

SQL SERVER es el producto emblema de motor de base de datos de Microsoft. Está generando una enorme cantidad de interés y excitación en el mercado, SQL SERVER se ha convertido en el principal motor de base de datos en la industria de cómputo para la plataforma Windows NT.

Los registros precio/rendimiento de SQL SERVER han permitido a muchas compañías tener la capacidad de un sistema de administración de base de datos relacionales (RDBMS, Relational Database Management System).

1.3.3.2 Características de SQL Server

SQL Server proporciona soporte para un conjunto de características que aportan las siguientes ventajas:

Facilidad de instalación, distribución y utilización

SQL Server incluye un conjunto de herramientas administrativas y de desarrollo que mejoran la capacidad para instalar, distribuir, administrar y utilizar SQL Server entre varios sitios.

Escalabilidad

Puede utilizarse el mismo motor de base de datos a través de plataformas que van desde equipos portátiles que ejecutan Microsoft Windows 95 ó 98 hasta grandes servidores con varios procesadores que ejecutan Windows NT, Enterprise Edition.

Almacenamiento de datos

SQL Server incluye herramientas para extraer y analizar datos resumidos para el proceso analítico en línea (OLAP, *Online Analytical Processing*). SQL Server incluye también herramientas para diseñar gráficamente las bases de datos y analizar los datos mediante preguntas en lenguaje normal.

1.3.3.3 SQL Server y el Modelo Cliente / Servidor.

Una aplicación cliente / servidor puede definirse como aquella que se divide en dos partes: una parte se ejecuta en un servidor y la otra se ejecuta en el cliente o estaciones de trabajo. El lado del servidor de la aplicación proporciona la seguridad, la tolerancia a fallas, el desempeño, la concurrencia y las copias de seguridad confiables. El lado del cliente proporciona la interfaz de usuario y puede contener informes, consultas y formularios vacíos. La idea consiste en tener lo mejor de ambos mundos aprovechando los dos y juntándolos.

SQL Server es la parte del servidor de la ecuación; 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.

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

1.3.3.3.2 VENTAJAS DEL SERVIDOR.

- Confiable.
- Concurrente.
- Bloqueo sofisticado.
- Tolerante a fallas.
- Hardware de alto desempeño.
- Control centralizado.

¹¹En la computación cliente7servidor, 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.

¹¹ UTC, Internet; <http://www.utem.cl/web/defbasedato.htm>. Último Acceso: jueves 18 de diciembre 2003.

1.3.4. RDBMS

1.3.4.1 ¿Que es un RDBMS?

CHARTE O. Francisco. Sostiene: “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*” (2002, pp. 540)

CAPITULO II

TRATAMIENTO DE LA INFORMACIÓN DE LA ACTIVIDAD ACADÉMICA DE LA CARRERA DE EDUCACIÓN SEMIPRESENCIAL EN EL CENTRO ASOCIADO LATACUNGA

2.1 LA CARRERA DE EDUCACIÓN SEMIPRESENCIAL EN LA UTC.

Reseña Histórica de la Carrera:

Como consecuencia de la evolución del pensamiento universal, y en los albores del inicio de un nuevo siglo, surge la imperiosa necesidad de optar por alternativas de educación Semipresencial.

Las instituciones formadoras de maestros no pueden permanecer indiferentes ante estos retos, siendo su compromiso ofertar programas educativos para todos los sectores sociales, bajo un paradigma de análisis crítico, como una propuesta alternativa que tenga eco en los potenciales usuarios. De ello se deriva la necesidad de ofertar especializaciones en docencia en el Centro Asociado: Latacunga a través de la modalidad semipresencial.

2.1.2 MISIÓN:

Formar profesionales en la docencia, en forma continua para los niveles primario; medio y áreas afines, con criterio abierto, pensamiento crítico, innovador y creativo, con elevado autoestima, valores éticos y morales, capaces de potenciar el desarrollo de la inteligencia y el pensamiento de los estudiantes; contribuir al crecimiento socio – económico, cultural y político del país, preservando su identidad, autonomía y libertad.

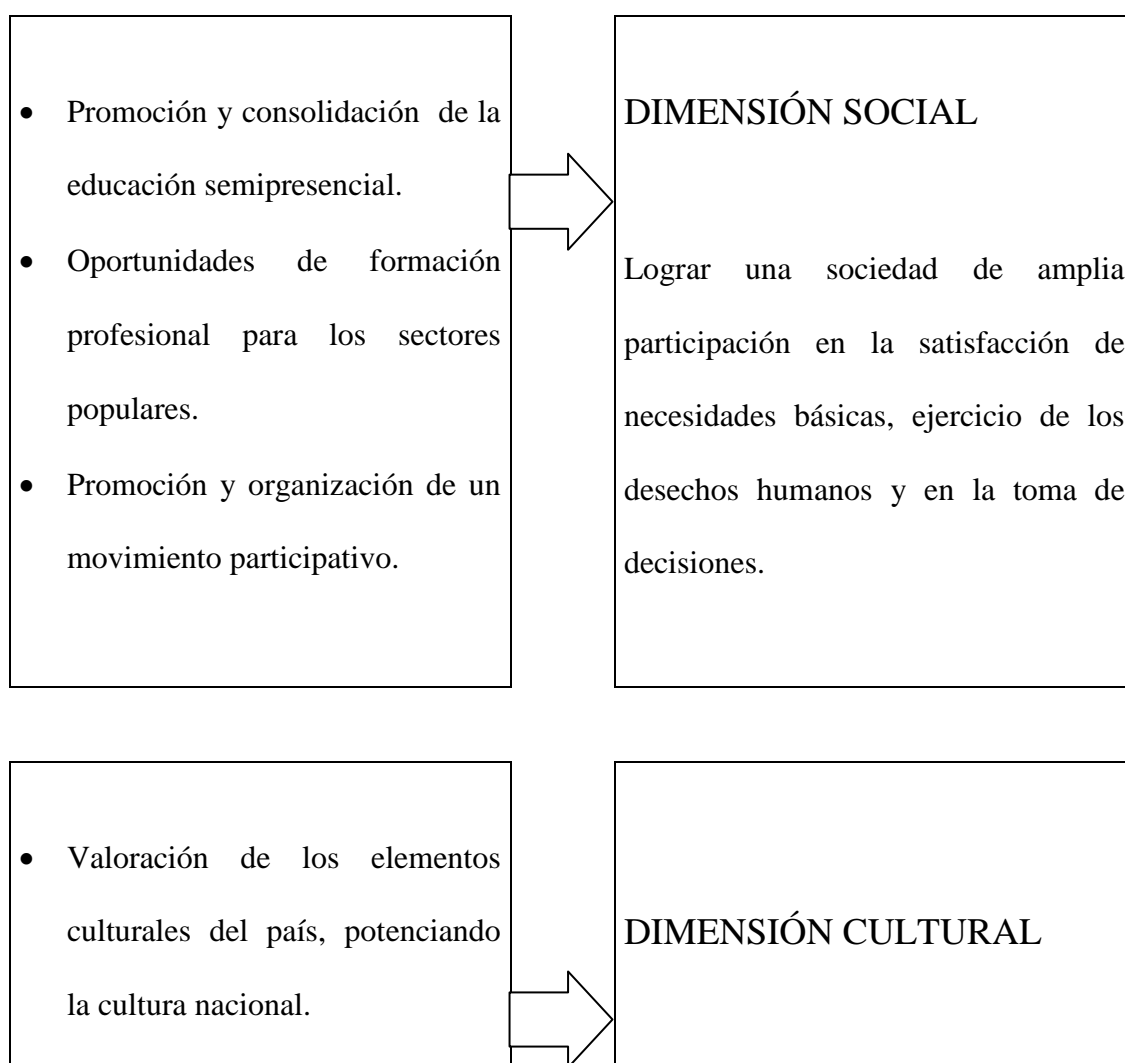
2.1.3 VISIÓN:

Liderar la formación profesional docente en servicio y preservicio de acuerdo con los avances científicos, tecnológicos y las necesidades de su contexto y las zonas que la requieran, para mejorar la calidad de la educación que promueva el desarrollo sustentable del país.

2.1.4 OBJETIVOS:

- Potenciar el desarrollo de la región, ampliando y dinamizando la cobertura de la educación universitaria, mediante la generación de oportunidades educativas.
- Facilitar los estudios universitarios considerando la demanda.
- Ofrecer una alternativa de educación semipresencial a quienes por diferentes motivos no pueden acceder a la educación presencial.

2.1.5 POLÍTICAS:



- Democratización de la cultura, ampliando la participación a los sectores marginales.
- Creación de tecnología educativa apropiada y asimilación crítica de la tecnología exterior.

Lograr una sociedad en la que sus miembros se identifiquen con la comunidad nacional, respetando la diversidad cultural, abriéndose a la cultura universal y fortaleciendo la autoformación e independencia nacionales.

2.1.6 ESTRATEGIAS:

- a) Constituir el Centro Asociado que desarrolle educación semipresencial para la formación profesional docente.
- b) Irradiar una formación docente que oriente la educación pre – primaria, primaria, media.
- c) Desarrollar investigación e innovación pedagógica para fundamentar el mejoramiento de la calidad de la educación.
- d) Diseñar y desarrollar el currículum de cada especialización con autonomía pedagógica, en relación con el contexto socio – económico local, sectorial y nacional.

2.1.7 METODOLOGÍA:

Los estudios dentro de cada especialidad se realizarán tomando como base el módulo estructurado para cada disciplina científica, acompañado de la correspondiente guía de estudio. Dicho módulo será desarrollado en el lugar de

residencia del estudiante; quien en caso de tener dificultades e inquietudes, acudirá al tutor vía telefónica, vía fax u otra. Además debe recopilar todas las dudas e inquietudes para ser absueltas en las tutorías.

ORIENTACIONES METODOLÓGICAS.

- En las tutorías se debe aplicar técnicas activas como: Foro, mesa redonda, debate, panel, simposio, círculos de estudio, etc.
- Para sistematizar los contenidos se aplicarán sistemas gráficos como: Menté afectos y mapas conceptuales.
- Ejercitar la autocrítica sobre la práctica de valores.
- Provocar experiencias vivenciales para fortalecer los valores de solidaridad, democracia y extensión universitaria.
- Realizar investigaciones de campo, bibliográfica y etnográfica.
- Elaborar proyectos educativos que posibiliten la Acción – Reflexión – Acción, en función de enriquecer su formación profesional.
- Se ampliarán textos, documentos de apoyo, vídeos, películas, láminas demostrativas, diapositivas y se priorizará el uso del INTERNET.

2.1.8 RECURSOS:

2.1.8.1 HUMANOS

- Director Académico de la Carrera
- Secretaria de la Carrera.
- Personal docente.

- Centro Asociado Latacunga

2.1.8.2 INFRAESTRUCTURA

- El Centro Asociado Latacunga funciona en el edificio de la Universidad con toda la implementación que posee la misma.

2.1.8.3 ECONÓMICOS

Los programas son autofinanciados en su totalidad por los alumnos.

2.1.9 REGLAMENTO PARA EL FUNCIONAMIENTO DEL ÁREA DE ESTUDIO A DISTANCIA

HONORABLE CONSEJO UNIVERSITARIO DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI.

CONSIDERANDO:

- Que en diferentes provincias del país solicitan a la Universidad Técnica de Cotopaxi y de manera particular de la Región Sur la puesta en marcha de Programas – Carrera con modalidad a distancia.
- Que es necesario ampliar la oferta de educación superior al sector social, mediante la modalidad a distancia en beneficio de grandes sectores humanos que por circunstancias de todo orden no han podido acceder a los estudios universitarios regulares.

- Que los estudios a distancia se operativizan a través de Centros Asociados; y dado su potencial de recursos humanos con experiencia y formación en educación a distancia con que cuenta la Universidad Técnica de Cotopaxi.

RESUELVE:

Crear el **ÁREA DE ESTUDIOS CON MODALIDAD A DISTANCIA** e implementar y organizar los Centros Asociados de: Loja, Latacunga y Riobamba, para la ejecución de los Programas Carrera a distancia, sin perjuicio de que en lo posterior se creen Centros en otras regiones del país.

Emitir el siguiente Reglamento para normar el funcionamiento de los Programas – Carrera a Distancia.

2.1.9.1 CAPITULO I (Artículos de la ley de la Carrera)

Art. 1. El Área de Estudios a Distancia estará bajo la dirección del Director Académico de Estudios a Distancia de la Universidad Técnica de Cotopaxi quien informará sobre su marcha ante el H. Consejo Universitario en forma periódica. Se conformará de todos los Centros Asociados que se crearen.

Art. 2. Los Centros Asociados estarán constituidos por un Coordinador; el mismo que será designado por el H. Consejo Universitario.

Art. 3. Es responsabilidad del Coordinador Académico.

- a) Planificar y proponer ante el H. Consejo Universitario de la U.T.C. diferentes Programas – Carrera.
- b) Elaborar y proponer ante el H. Consejo Universitario de la U.T.C. el calendario de actividades Académicas.
- c) Presentar para su aprobación ante el H. Consejo Universitario de la U.T.C., los planes y programas de estudio así como sus modalidades de trabajo académico científico.
- d) Sugerir ante el H. Consejo Universitario la aprobación de los horarios de estudio; y apertura y cierre de los diferentes períodos de matrículas.
- e) Conocer y resolver de acuerdo a la normatividad del Centro, sobre los problemas que se presentaren en el desarrollo de las actividades académico – administrativa de los programas que administra el Centro Asociado.
- f) Responsabilizarse por la buena marcha académico – administrativa de los programas del Centro.
- g) Remitir al menos dos veces por ciclo, ante el Rector de la U.T.C.; Informe sobre el funcionamiento académico administrativo de los programas carrera.

- h) Promover la evaluación y seguimiento académica para garantizar la calidad y excelencia de los estudios.
- i) Ordenar a Secretaria del Centro la recepción de la documentación de matrículas en sujeción con las disposiciones legales de la U.T.C. y del presente instructivo informará de este particular al Rector de la Universidad.
- j) El Coordinador del Centro Asociado será el Representante Legal de la U.T.C. dentro del espacio académico que incumbe a los estudios a distancia del referido centro. Debiendo además asumir otras funciones que el H.C.U., el Rector u otra autoridad universitaria le delegaren.
- k) Propiciar la firma de convenios entre la U.T.C. y otras instituciones de desarrollo, en miras al fortalecimiento de la labor académica y del desarrollo de las regiones del país.
- l) Organizar programas de docencia, investigación y extensión tendientes a vincular la universidad con la sociedad.
- m) Asumir la función para contratar, remover o sancionar al personal docente y administrativo que se requiere para la buena marcha de los programas.
- n) Coordinar y administrar el trabajo docente – educativo, apoyándose en la Secretaría del Centro y las Secretarías de coordinación de los cantones bajo su responsabilidad.

- o) Coordinar para que los alumnos cumplan oportunamente sus obligaciones económicas con la universidad.
- p) Propiciar la entrega oportuna a los alumnos del material de estudio.
- q) Facilitar la labor de los docentes y alumnos, así como aceptar los trabajos y calificaciones que surjan de las evaluaciones de los docentes.
- r) Recibir los informes de avance y culminación del módulo y remitirlos al Vicerrector de la U.T.C. para su aprobación y honorarios al docente de acuerdo al respectivo contrato.

Art. 4. Son funciones de la Secretaria General del Centro:

- a) Dar información a docentes y alumnos, respecto a las asistencias, calificaciones, inicio de matrículas y otros aspectos del proceso docente educativo.
- b) Llevar registro de asistencia de los administrativos, docentes y alumnos del Centro Asociado.
- c) Contribuir con el Coordinador del Centro en todo lo que sea administración y seguimiento de los programas – carrera.
- d) Llevar registros de archivo y comunicaciones desde y hacia el Centro Asociado.
- e) Atender en forma continua y durante 8 horas diarias en las diferentes actividades que se emanen desde y hacia el Centro Asociado.

- f) Distribuir hacia las secretarías de coordinación de los cantones del área de influencia del Centro, los materiales requeridos para el estudio por parte de los alumnos.
- g) Contribuir en la producción de materiales requeridos en el proceso docente educativo de los módulos.
- h) Receptar la documentación de matrículas de acuerdo a las directrices emitidas por las instancias superiores de la Universidad.
- i) Elaborar los contratos a celebrarse entre la Universidad y los docentes del respectivo Centro.

2.1.9.2 DE LOS DOCENTES.

Art. 5. Los docentes del Centro Asociado, serán contratados por el Sistema Contrato de Servicios Personales para el tiempo que se requiera sus servicios y sus honorarios serán de acuerdo a lo presupuestado en cada uno de los ciclos. Para efectos de honorarios se considerará cinco meses de labor por cada módulo y en el caso del curso taller el tiempo que dure éste.

Art. 6. El pago a los docentes de cada módulo se hará, el 30% como anticipo, otro 30% luego del primer informe del Director y el 40% restante al finalizar el mismo. En el caso de los docentes del curso o taller de apoyo el pago se hará una vez que haya culminado el respectivo evento académico.

Art. 7. Para ser docente contratado del Centro Asociado se requiere:

- a) Tener Título Académico.

- b) Haber aprobado el Curso Básico de Docencia Universitaria e Investigación Educativa de Modalidad a Distancia.

Art. 8. Son atribuciones y deberes de los docentes:

- a) Participar en la planificación, ejecución y evaluación del Programa – Carrera.
- b) Asistir puntualmente a las labores y sesiones de trabajo y actos programados por el Centro Asociado y la Universidad en general.
- c) Al término de cada fase el docente presentar ante la Secretaría del Centro Asociado un informe de actividades donde se haga constar los trabajos de evaluación con sus respectivas calificaciones.
- d) Integrar las comisiones que le fueren designadas.
- e) Los demás que determine la Universidad.

Art. 9. Los docentes podrán ser removidos de sus cargos por el Rector de la U.T.C., previo informe del Coordinador del Centro Asociado de acuerdo a lo estipulado en el respectivo contrato de servicios personales.

2.1.9.3 DE LOS ESTUDIANTES

Art. 10. Quienes hayan obtenido legalmente la matrícula en el Centro Asociado, serán considerados estudiantes de la U.T.C., por lo tanto son acreedores a sus deberes y derechos.

Art. 11. Asistir al 90% las actividades desarrolladas con motivo de sus estudios en cada Programa – Carrera así como cumplir con los requerimientos de la planificación curricular.

2.1.9.4 DE LA EVALUACIÓN Y ACREDITACIÓN

Art. 12. La evaluación versará sobre todos los aspectos que influyen en el proceso enseñanza aprendizaje; y, la acreditación será en base a productos acreditables que resulten de los siguientes parámetros:

- | | |
|-----------------------------------|-----|
| a) Investigación | 40% |
| b) Guía de Estudios | 30% |
| c) Trabajos individuales grupales | 10% |
| d) Pruebas de recapitulación | 20% |

Art. 13. El ciclo de estudio durará 18 semanas, tiempo en el cual se desarrollarán los módulos y cursos de apoyo. Durante el desarrollo del módulo se considerará dos fases para la acreditación, la primera el término del segundo mes y la segunda en la última sesión presencial. La nota final de acreditación será el promedio que resulte de ambas fases, debiendo reunir para su aprobación por lo menos el 80% de la calificación total.

(Art.13) reformado según resolución del H. Consejo Universitario del miércoles 21 de julio de 1999, en sesión ordinaria.

Art. 14. En caso de que el estudiante no lograra alcanzar la nota requerida para la acreditación, previo la adquisición de un derecho equivalente al 20% de un salario mínimo vital, podrá solicitar al docente se le envíe tareas que le permitan nuevas oportunidades de aprendizaje y su consecuente mejoramiento de nota, las cuales deberán ser entregados hasta antes de iniciar el siguiente ciclo.

Art. 15. Si un estudiante no se presentare a las pruebas de recapitulación puede hacerlo en el plazo máximo de cinco días una vez que haya cancelado el derecho correspondiente.

2.1.9.5 DE LA GRADUACIÓN Y CERTIFICACIÓN

Art. 16. Para graduarse en la Universidad Técnica de Cotopaxi, se seguirán los procedimientos establecidos en el Reglamento respectivo.

Art. 17. Las certificaciones de asistencia a clases de parte de los estudiantes, serán expedidas por el Secretario (a) del Centro Asociado.

Art. 18. La expedición de títulos, certificados de egresamiento, matrícula y aprobación de módulos será emitidos por las diferentes instancias de la U.T.C., de acuerdo a las leyes y reglamentos de la Universidad en base de la información debidamente sellada y firmada por el Secretario y Coordinador que proporcione las Secretarías de los Centros Asociados.

2.1.9.6 DE LOS COSTOS Y FINANCIAMIENTO

Art. 19. Los costos de matrícula y pensiones, serán fijados por el H. Consejo Universitario de la U.T.C.

Art. 20. Todos los programas que desarrolle el Centro Asociado se basarán en el principio de autosostenimiento.

Art. 21. El personal docente y administrativo que se requiera, así como sus horarios, será determinado por el H. Consejo Universitario previa solicitud del Coordinador del Centro Asociado.

Art. 22. Los gastos de operación y desarrollo del Centro Asociado, así como los honorarios del personal docente y administrativo serán determinados en el presupuesto, el mismo que será formulado para cada ciclo de estudios.

Art. 23. El estudiante pagará matrículas y pensiones, con lo cual tiene el derecho de recibirlos servicios académicos y administrativos así como el material de estudio que se requiera.

Art. 24. Los valores por pago de matrícula, pensiones u otros, se depositarán en las cuentas de la Universidad e irán orientados a financiar el diseño y publicación de

material de estudio así como arriendo de locales. Los valores procedentes de las pensiones ingresarán en tesorería de la U.T.C. y servirán para el pago del personal docente y administrativo, y otros egresos, del Centro Asociado.

Art. 25. Los valores por pago de derecho de laboratorio y gabinete, se utilizarán en el pago de estos servicios a personas o instituciones que los brindaren.

Lo no previsto en el presente Reglamento se regirá por el Reglamento General de la Universidad, el Estatuto y la Ley de Universidades y Escuelas Politécnicas.

Dado en Latacunga, en la sala de sesiones del H. Consejo Universitario de la Universidad Técnica de Cotopaxi a los nueve días del mes de Abril de mil novecientos noventa y siete.

2.2 EL MANEJO DE LA INFORMACIÓN EN LA CARRERA DE EDUCACIÓN SEMIPRESENCIAL, EN LA ACTUALIDAD

CENTRO ASOCIADO LATACUNGA (MATRIZ)

ESPECIALIDAD: EDUCACION BASICA

DURACIÓN: Ocho Ciclos

GRADUACIÓN: Tesis de Grado y su Defensa

TITULO A OBTENER: Licenciado (a) en Ciencias de la Educación, Especialidad Educación Básica.

ESTRUCTURA ACADEMICA

2.2.1 OBJETIVOS DE LA CARRERA:

- Contribuir con el país mediante la dotación de recursos humanos preparados para insertarse en la práctica de Educación Básica.
- Potenciar la docencia en el nivel básico a través de la preparación de quienes estando en esa práctica no poseen la respectiva profesionalización.
- Formar Profesionales en Educación Básica cuyo perfil está dado por caracteres de INVESTIGAR LO QUE SE ENSEÑA Y ENSEÑAR LO QUE SE INVESTIGA logrando así coherencia con las necesidades y retos del desarrollo del país.
- Dar facilidades de profesionalización en Educación Básica a personas que por diferentes situaciones no pueden abandonar su lugar de trabajo para adquirir dicha formación en forma presencial.

2.2.2 PERFIL PROFESIONAL:

El profesional en Educación Básica tendrá las siguientes características:

- 1.- Preparado con suficientes elementos filosóficos, epistemológicos, teóricos, metodológico- técnicos requeridos en la investigación social y educativa, con lo cual se incorporará a su práctica en forma crítica, creativa y propositiva.

2.- Con alta formación disciplinaria en las diferentes áreas implicadas en la Educación Básica capaz que ello les permita una eficiente coordinación del proceso enseñanza – aprendizaje en lo que a formación disciplinar se refiere.

3.- Con conocimientos, habilidades y destrezas para propiciar investigaciones en los diferentes aspectos de la Educación Básica, a efectos de que el profesionalista pueda adaptar, crear y proponer nuevas formas de administración, organización curricular y de enseñanza – aprendizaje acorde con la realidad ecuatoriana y del entorno del Centro Educativo donde labore.

4.- Capaces de actuar corpórea – estética y oralmente en los diferentes aspectos de la Educación Básica en particular y de la diversidad social, cultura y étnica ecuatoriana en General.

2.2.3 REQUISITOS: (PRIMER CICLO)

- Título original y fotocopia
- Acta refrendada y fotocopia
- Certificado de estar en servicio activo (actualizado)

Fiscales: De la Dirección Provincial de Educación

Particulares: Director o Rector de la institución respectiva (acreditar 3 años de servicio interrumpido).

- 4 fotos a color, tamaño carné
- Fotocopia de documentos personales: cédula de ciudadanía, certificado de votación, cédula militar (hombres).
- Pago de los derechos arancelarios.
- Récord policial (actualizado).

A PARTIR DE SEGUNDO CICLO

- Haber pagado las pensiones
- Fotocopia de los documentos personales
- 2 fotos tamaño carné

2.3 CRITERIOS DE LOS DOCENTES

De acuerdo a los instrumentos de investigación utilizados como son las encuestas que se realizaron a la comunidad universitaria de la Carrera Semipresencial Especialidad “Educación Básica” del Centro Asociado Latacunga se obtuvieron los resultados que a continuación se detalla:

PREGUNTA N° 1

¿Cómo calificaría la agilidad de la información y comunicación que brinda la secretaria de la carrera?

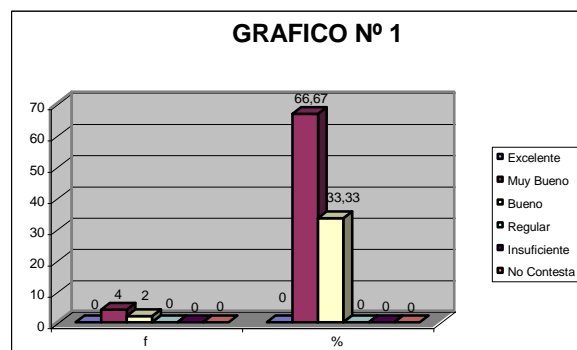
TABLA N° 1

ALTERNATIVAS	f	%
Excelente	0	0
Muy Bueno	4	66.67
Bueno	2	33.33
Regular	0	0

Insuficiente	0	0
No Contesta	0	0
TOTAL	6	100.00

Fuente: Encuesta a Profesores.

Elaboración: Integrantes de la Tesis.



De un total de 6 encuestados determinamos los siguientes porcentajes expresados de la siguiente manera 4 maestros que equivale al 67% califican que la agilidad de la secretaria de la carrera es muy buena, 2 maestros que corresponde al 34% determinan que es bueno, en virtud de lo encuestado los autores determinan que:

- Al momento de realizar una gestión o trámites en esta dependencia se realiza en un marco de agilidad y prontitud.
- En virtud del número de estudiante se avizora dificultades en la atención a los estudiantes en general.

PREGUNTA N° 2

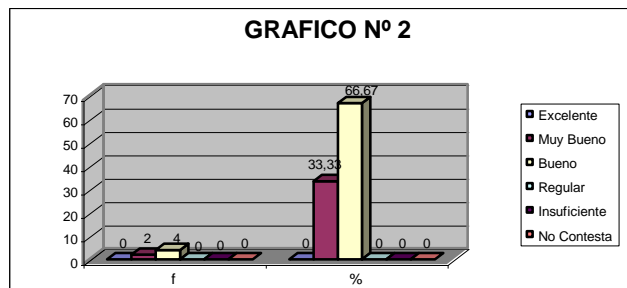
¿Cómo calificaría la agilidad de la información y comunicación que brinda el Director Académico de la carrera?

TABLA N° 2

ALTERNATIVAS	f	%
Excelente	0	0
Muy Bueno	2	33.33
Bueno	4	66.67
Regular	0	0
Insuficiente	0	0
No Contesta	0	0
2.4 TOTAL	6	100.00

Fuente: Encuesta a Profesores.

Elaboración: Integrantes de la Tesis.



Según la tabulación hemos conocidos que 4 maestros equivalentes al 67% consideran que la comunicación y agilidad que dispensa el Director de carrera es bueno, y 2 maestros con un porcentaje del 33% expresan que es muy buena debido a las siguientes aseveraciones por los que autores consideran:

- Falta de socialización en algunos temas que se debería comunicar con la oportunidad del caso.
- No entrega suficiente tiempo a las actividades de la carrera incluyendo la puntualidad en las citas programadas.

PREGUNTA N° 3

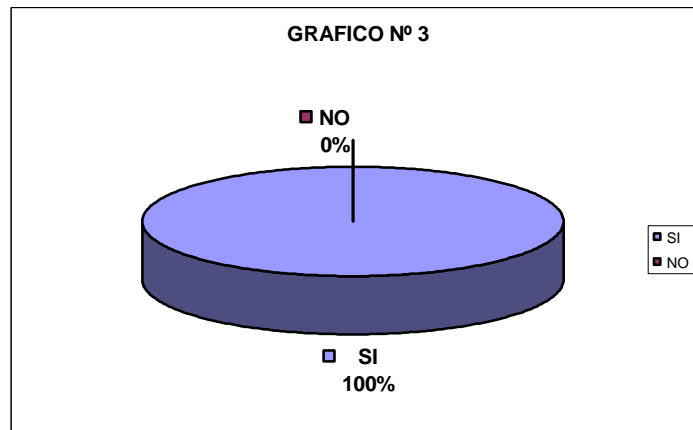
¿Considera que la información que brinda la administración de la Carrera debe ser más eficiente?

TABLA N° 3

ALTERNATIVAS	f	%
SI	6	100.00
NO	0	0
TOTAL	6	100.00

Fuente: Encuesta a Profesores.

Elaboración: Integrantes de la Tesis.



De acuerdo a los datos tabulados los encuestados determinan de un total 6 maestros equivalente al 100%, consideran que si debe ser más eficiente la información de la administración de la carrera; en virtud de los resultados los autores creen que:

- Siempre será importante que las dependencias de la carrera semipresencial ofrezcan un mejor servicio a los estudiantes.
- Que se mantenga puntualidad en el horario de trabajo ya que esto permitirá dilucidar algunas observaciones y preguntas que mantienen tanto estudiantes como maestros.

PREGUNTA N° 4

¿En que aspectos considera que debe mejorar la Administración de la Carrera, con relación a la entrega de información?

Con los datos obtenidos los autores determinaron el siguiente análisis:

- La casi nula planificación, dirección y organización por parte de la Administración de la Carrera.
- Que se debe automatizar la información de la carrera lo cual permitiría mayor rapidez en el flujo de los petitorios tanto de maestros como estudiantes.

PREGUNTA N° 5

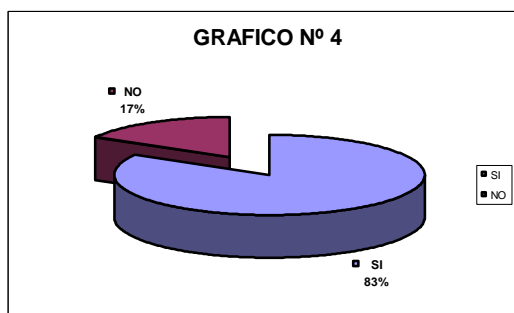
Si la UTC crea una Base de Datos con la información de la Actividad Académica de la Carrera de Educación Semipresencial en una red local (Informática), usted hiciera uso de ella para información de calificaciones, tareas de investigación, tanto a estudiantes como maestros, etc.

TABLA N° 4

ALTERNATIVAS	f	%
SI	5	83.33
NO	1	16.67
TOTAL	6	100.00

Fuente: Encuesta a Profesores.

Elaboración: Integrantes de la Tesis.



De los resultados obtenidos en esta pregunta, se puede destacar que: casi la totalidad de docentes que contestaron la misma, tienen una visión muy clara del aprovechamiento de este recurso (Base de Datos) como innovación tecnológica y como medio de difusión informativa para los estudiantes.

PREGUNTA N° 6

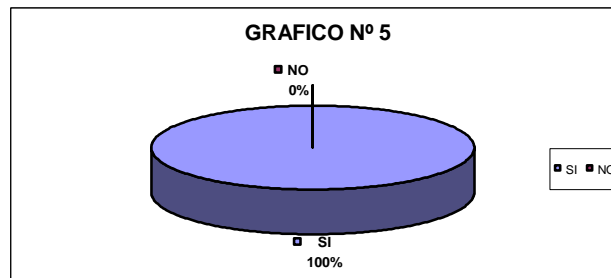
Considera que el uso de la tecnología informática y la creación de un Sistema Académico permitirán superar las dificultades de comunicación e información a los estudiantes de la Carrera.

TABLA N° 5

ALTERNATIVAS	f	%
SI	6	100.00
NO	0	0
TOTAL	6	100.00

Fuente: Encuesta a Profesores.

Elaboración: Integrantes de la Tesis.



Al preguntar a los maestros si la creación de un sistema académico permitiría superar las dificultades de comunicación e información, 6 maestros equivalentes al 100% responden que si les gusta que se realice este sistema informático, por lo que los investigadores acuerdan que:

- La información de la carrera de manejaría en forma más rápida y actualizada tanto para los estudiantes, maestros y autoridades y al mismo tiempo se aprovecharía el gran avance de la tecnología informática y comunicación.

PREGUNTA 7.

Qué información, de entre las que entrega el docente, considera que debe constar en el Sistema Académico.

Una vez analizado los resultados de esta pregunta abierta, los autores han llegado a conocer que en forma general, la información que los docentes desean que se implante en el Sistema Académico.

- Las calificaciones.
- Los módulos para los estudiantes.
- Planificaciones curriculares.
- Plan Analítico
- Mallas curriculares
- Trabajos de Investigación.
- Horario de las tutorías.
- Innovaciones o aportes metodológicos e investigativos.

PREGUNTA N° 8.

Que información de entre la que entrega la Administración de la Carrera, considera que debe constar en el Sistema Académico.

De las respuestas vertidas en esta pregunta, los autores pueden resaltar que la información que debe figurar en el Sistema Académico es:

- Calendario Académico.
- Nómina de los estudiantes pertenecientes a cada ciclo.
- Pensum de estudios de los ciclos.
- Pases de ciclo.

2.4 CRITERIOS DE LOS ESTUDIANTES.

PREGUNTA N° 1

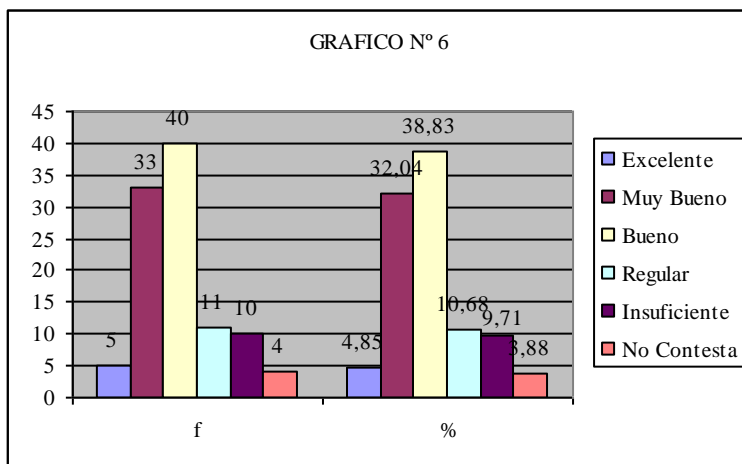
¿Cómo calificaría la agilidad de la información y comunicación que brinda la secretaria de la Carrera?

TABLA N° 6

ALTERNATIVAS	f	%
Excelente	5	4.85
Muy Bueno	33	32.04
Bueno	40	38.83
Regular	11	10.68
Insuficiente	10	9.71
No Contesta	4	3.88
TOTAL	103	100.00

Fuente: Encuesta a Estudiantes

Elaboración: Integrantes de la Tesis.



Al investigar acerca de la agilidad de la información y comunicación de la secretaria se obtuvo que: 5 estudiantes equivalentes al 4.85% manifiestan que es excelente, 33 estudiantes equivalentes al 32.04% expresan que es muy bueno, 40 estudiantes que representan el 38.83% se pronuncian que es bueno, 11 estudiantes que representan el

10.68% indican que la agilidad es regular, 10 estudiantes que equivalen al 9.71% manifiestan que es insuficiente además de 4 estudiantes que representan el 3.88% no tienen ningún pronunciamiento acerca de las opciones antes encuestadas; de acuerdo a la opinión de la mayoría los autores determinan que la comunicación e información de la secretaria de Carrera es bueno debido a:

- Hasta el momento la gestión y trámites desarrollados en esta dependencia es regular por parte de las autoridades.
- Existen dificultades por el número de alumnos.
- No facilitan documentación sugerida y en muchas veces no se encuentran en el puesto de trabajo.
- No existe una comunicación y coordinación entre los estudiantes y la secretaria y viceversa motivo por el cual suceden varios problemas.
- Se evidencia también la falta de relaciones humana y ética profesional.
- No cumple con el horario establecido por las autoridades de la universidad.

PREGUNTA N° 2

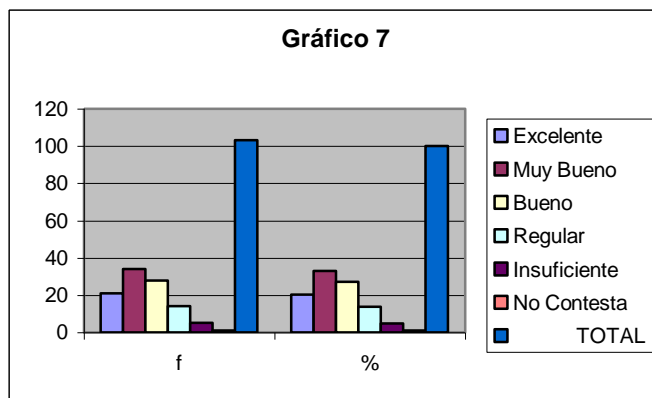
¿Cómo calificaría la agilidad de la información y comunicación que brinda el Director Académico de la Carrera?

TABLA N° 7

ALTERNATIVAS	F	%
Excelente	21	20.39
Muy Bueno	34	33.01
Bueno	28	27.18
Regular	14	13.59
Insuficiente	5	4.85
No Contesta	1	0.97
TOTAL	103	100.00

Fuente: Encuesta a estudiantes.

Elaboración: Integrantes de la Tesis.



En lo referente a la agilidad del Director Académico 21 estudiantes, equivalente al 20.39% expresan que la agilidad de la información del Director es excelente, 34 correspondiente al 33.01% manifiestan que es muy bueno, 28 alumnos equivalentes al

27.18% indican que es bueno, 14 alumnos que corresponde al 13.59 coinciden en que es regular, 5 estudiantes equivalentes al 4.85 consideran insuficiente y 1 estudiante que corresponde al 0.97% no responde. Los autores frente a los resultados expresan que la atención que dispensa el Director Académico es muy buena debido a los siguientes razonamientos por parte de los autores:

- Porque la información y comunicación que es necesaria para los estudiantes sea oportuna y clara.
- Ocasionalmente se lo encuentra en la oficina por lo que se hace difícil mantener una comunicación personalmente.
- Falta comunicar sobre algún evento social y emite recomendaciones.
- En las oportunidades que lo hemos visitado nos ayuda a solucionar los problemas y necesidades del estudiante, es una persona amable y capaz.
- Está pendiente de tramitar y agilizar todas las actividades planificadas en beneficio de la carrera y de la institución.
- Presenta deficiencia en comunicar las resoluciones tomadas en Consejo Académico y direcciones de carrera.

PREGUNTA N° 3

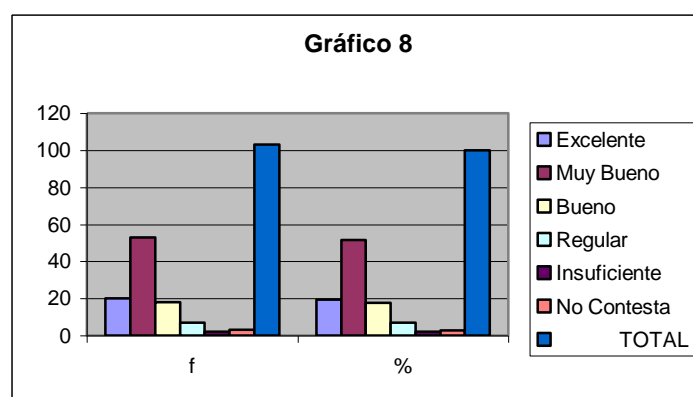
¿Cómo calificaría la agilidad de la información y comunicación que brindan los docentes de la carrera?

TABLA N° 8

ALTERNATIVAS	f	%
Excelente	20	19.42
Muy Bueno	53	51.46
Bueno	18	17.48
Regular	7	6.80
Insuficiente	2	1.94
No Contesta	3	2.91
TOTAL	103	100.00

Fuente: Encuesta a estudiantes.

Elaboración: Integrantes de la Tesis.



Por medio de esta encuesta determinamos que 20 estudiantes equivalentes al 19.42%, indican que es excelente la agilidad, comunicación e información de los docentes, 53 equivalente al 51.46% responden que la agilidad de los docentes es muy buena, 18 estudiantes correspondientes al 17.48 expresan que es buena, 7 estudiantes que corresponde al 6.80% coinciden que agilidad en la información de los docentes de la Carrera es regular, 2 estudiantes que significa el 1.94% indican que es insuficiente y 3

estudiantes con un porcentaje del 2.91% no dan respuesta. La información nos permite comprender que la mayoría de docentes proporcionan agilidad en la información y comunicación para los estudiantes. De acuerdo a un análisis por parte de los autores hemos considerado lo siguiente:

- Transmiten los conocimientos de forma clara y permiten desarrollar exposiciones de los alumnos para completar con un refuerzo final.
- Podemos expresar que existen profesores excelentes pero también algunos a los cuales no se entienden las tutorías.
- Tiene gran calidad humana y lo que es mas entienden y comparten algunos problemas que a los estudiantes nos ocurren.
- Tienen un amplio conocimiento de la materia la cual dictan y están dispuestos aceptar nuestras sugerencias en el aspecto enseñanza-aprendizaje.
- Antes de dictar su hora clase algunos preparan la clase, planifican y tienen la forma de evaluar aplicando una pedagogía acorde al tema.

PREGUNTA N° 4

Considera que la información que brinda la administración de la Carrera debe ser más eficiente.

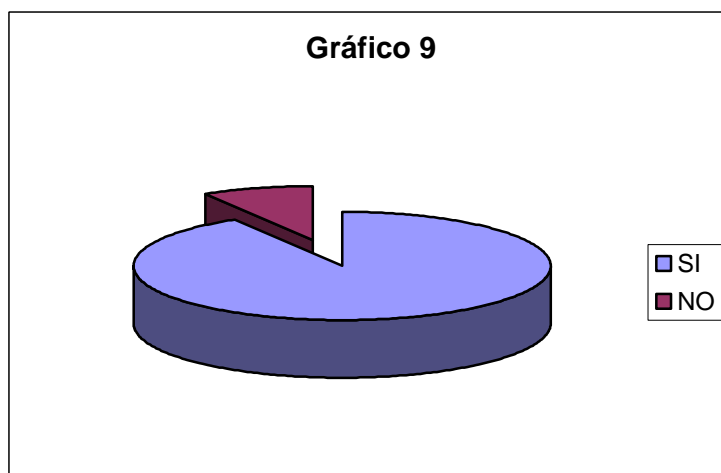
TABLA N° 9

2.5 ALTERNATIVAS	f	%
SI	94	91.26
NO	9	8.74

TOTAL	103	100.00
--------------	------------	---------------

Fuente: Encuesta a estudiantes.

Elaboración: Integrantes de la Tesis.



En las encuestas realizadas, 94 estudiantes equivalentes al 91% contestan que si debe ser más eficiente la información de la Administración de la Carrera y 9 estudiantes equivalentes al 9% indican que no debe ser más eficiente por autores creen que:

- Existe una desinformación para con los estudiantes de Educación a Distancia por parte de las autoridades de la carrera.
- No dan cumplimiento a los calendarios académicos que nos comunican.
- Exista mas flexibilidad en los horarios de matriculas y eventos que realiza la carrera porque nuestra residencia es lejana.

- La información que nos proporcionan no es clara y precisa.
- Se debería automatizar y realizar una planificación estratégica con visión de futuro.
- Es palpable la desorganización de nuestra documentación lo cual provoca confusiones y en muchas ocasiones perdidas.
- En lo posible elaborar un horario especial los días de tutorías en vista que los estudiantes asisten y disponen del tiempo para informarse sobre algún acontecimiento.

PREGUNTA N° 5

Si la UTC crea un sistema con la información de la Actividad Académica de la Carrera de Educación Semipresencial en una red local (informática), usted hiciere uso de elle para informarse de los requisitos de matriculas, pases de ciclo, calificaciones, tareas de investigación, etc.

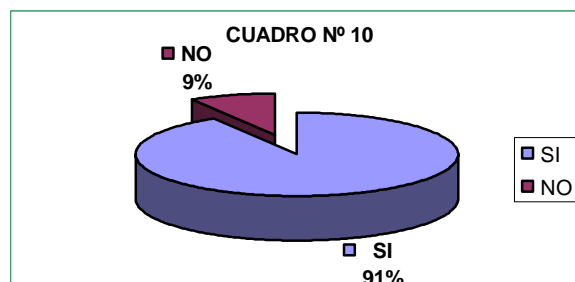
TABLA N° 10

ALTERNATIVAS	f	%
SI	94	91.26
NO	9	8.74

TOTAL	103	100.00
--------------	------------	---------------

Fuente: Encuesta a estudiantes.

Elaboración: Integrantes de la Tesis.



Según la tabulación hemos conocido que 94 estudiantes equivalentes al 91%, consideran que les gustaría tener un sistema con la información de la Carrera y 9 que corresponden al 9% indican que no; sobre la base de estos datos los tesisistas consideran:

- Sería de enorme facilidad y veracidad en la información de nuestra carrera y estaría al alcance en cualquier lugar dentro del campus universitario.
- Por que el avance tecnológico se haría presente en nuestra carrera y esto evitaría al estudiante hacer largas colas en busca de una información.
- Sería más fácil, útil y práctico y a lo mejor ya no necesitaríamos estar preguntando a los empleados administrativos.
- Se constituiría en una fuente de consulta, ahorro de tiempo y mucho más eficiente.

- Es una brillante idea ya que en toda institución la información se maneja mediante las computadoras.
- Es una información que permanece todos los días, y no sería indispensable asistir a una hora exacta para preguntar al personal administrativo.

PREGUNTA N° 6

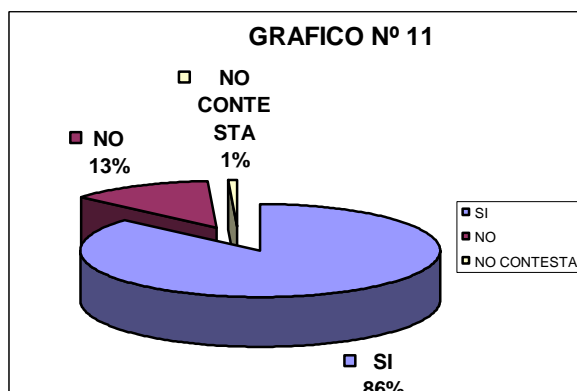
Considera que el uso de la tecnología informática y la creación de un Sistema Académico (informática) permitirán superar las dificultades de comunicación e información en la Carrera.

TABLA N° 11

ALTERNATIVAS	F	%
SI	89	86.41
NO	13	12.62
NO CONTESTA	1	0.97
TOTAL	103	100.00

Fuente: Encuesta a estudiantes.

Elaboración: Integrantes de la Tesis.



De los 103 estudiantes encuestados, 89 equivalentes al 86% consideran que la creación de un Sistema Informático permitiría superar muchas dificultades de comunicación e información, 13 equivalente al 13% responden que no y 1 correspondiente al 1% no responde a la pregunta.

- Sería de gran valía porque para solicitar información ya no tendríamos que acudir donde la secretaria.
- Por medio de este sistema se daría a conocer nuestra carrera y lo que es mas la propia universidad, en tal virtud existiría mayor afluencia de estudiantes.
- Este sistema obligaría a los maestros a que emitan las notas en la fecha determinada lo cual a nosotros como estudiantes nos ahorraría tiempo ya que la mayoría de nosotros trabajamos en lugares apartados de la ciudad.
- Sería una herramienta muy práctica y dinámica de gran ayuda tanto para administrativos, docentes y estudiantes.
- Indispensable para superar todas las dificultades que se presentan en la actualidad, dejar de lado lo tradicional ya es hora de modernizarnos.

CAPÍTULO III

**DISEÑO DE SOFTWARE PARA EL MANEJO DE BASE DE DATOS CON
INFORMACIÓN DE LA ACTIVIDAD ACADÉMICA DE LA CARRERA DE
EDUCACIÓN SEMIPRESENCIAL EN EL CENTRO ASOCIADO
LATACUNGA DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI**

3.1. Modelo de construcción del software

3.1.1. Modelo de Fases.

3.1.1.1. Análisis

3.1.1.1.1. Fase de Planificación y Especificación de Requisito

El sistema “**EBCAL UTC**” será desarrollado bajo la notación UML, ya que constituye un estándar en relación al desarrollo de software orientada a objetos.

UML permite abarcar todos y cada de las fases del ciclo de vida de un software, desde el análisis de requerimientos hasta la implantación y pruebas del mismo. Esta metodología introduce los Casos de Uso, una poderosa herramienta para reducir los riesgos en la definición de requerimientos de sistemas nuevos. Los Casos de Uso sirven como columna vertebral del proceso de desarrollo de aplicaciones y tienen como objetivo garantizar que los resultados se ajusten completamente a las expectativas de los usuarios finales.

3.1.1.1.1.1. Planificación

3.1.1.1.1.1.1. Propósito

Mediante esta fase se busca conocer y describir de una forma general el ámbito del sistema, alcance, perspectiva, funciones y sus requisitos, los mismos que se relaciona con los procesos académicos que realiza la Especialidad de Educación Semipresencial del Centro Asociado Latacunga.

3.1.1.1.1.2. Ámbito del Sistema

De acuerdo a un estudio de factibilidad realizado en la Especialidad de Educación Semipresencial del Centro Asociado Latacunga, se ha creído conveniente llevar a efecto el desarrollo de un software confiable que proporcione una información rápida y eficiente, utilizando los avances tecnológicos de última generación.

Al existir un número considerable de estudiantes, se torna necesario contar con un apoyo informático que provea una incondicional e invaluable ayuda en la Secretaría de la Especialidad de Educación Semipresencial del Centro Asociado Latacunga.

3.1.1.1.1.3. Alcance del Sistema

El sistema “**EBCAL UTC**” constituye un aporte importante para este sector de la Universidad Técnica de Cotopaxi. Su alcance ha sido establecido de acuerdo a la población de estudiantes, docentes y personal administrativo que pertenecen a la Especialidad de Educación

Semipresencial del Centro Asociado Latacunga, lo cual se detalla en la tabla 12 y tabla 13:

Tabla 12. Población de Estudiantes de la Carrera Semipresencial

CICLO	NÚMERO DE ESTUDIANTES
SEGUNDO	8
TERCERO	11
CUARTO	26
QUINTO	31
SEXTO	31
Total:	107

Fuente: Los Autores

Tabla 13. Población de autoridades, administrativos y docentes de la Carrera Semipresencial

AUTORIDADES	1
ADMINISTRATIVOS	1

DOCENTES	6
Total:	8

Fuente: Los Autores

3.1.1.1.1.4. Definición de Usuarios

Tabla 14. Usuarios del Sistema

USUARIO	ACTIVIDAD
Secretario/@	Gestiona las actividades que se cumplen en la Secretaría de Especialidad.
Alumno/@	Solicita la realización de una operación al sistema.
Docente	Solicita la realización de una operación al sistema.
Director/@ de Especialidad	Se encarga de la administración de la Especialidad

Fuente: Los Autores.

3.1.1.1.2. Descripción General

Se pretende realizar una descripción de las funciones, características y limitantes que deberán ser tomados muy en cuenta en el desarrollo del sistema.

3.1.1.1.2.1. Funciones del Sistema

El sistema “**EBCAL UTC**” acogerá las siguientes funciones, las mismas que han sido consideradas mediante un análisis del flujo de actividades que se desarrollan en la Secretaría de la Especialidad de Educación Semipresencial del Centro Asociado Latacunga:

- Gestión de Materias.
- Gestión de Docentes.
- Gestión de Alumnos.
- Gestión de Notas y Faltas.
- Gestión de Historial.
- Gestión de Egresados

En la tabla 15 se especifica los procesos que cumple cada una de las funciones de gestión detalladas anteriormente:

Tabla 15. Gestión de procesos

GESTIÓN	PROCESOS			
	AL TAS	ESTADO	CAMBIOS	REPORTES
Materias			✓	✓
Docentes	✓	✓	✓	✓
Alumnos	✓	✓	✓	✓
Notas/Faltas	✓	✓	✓	✓
Historial	✓			✓
Cierre del Sistema	✓		✓	✓
Egresados	✓			✓

Fuente: Los Autores.

3.1.1.1.3. Requisitos Específicos

3.1.1.1.3.1. Requisitos Funcionales

3.1.1.1.3.1.1. Gestión de Materias

Los procesos que el sistema podrá realizar son los siguientes:

- Modificar el nombre de las materias.
- Reporte de las materias de manera general.
- Reporte de materias por cada ciclo.

3.1.1.1.3.1.2. Gestión de Docentes

Los procesos que el sistema podrá realizar son los siguientes:

- Ingreso de nuevos docentes.
- Paso a inactividad de docentes.
- Modificación de datos del docente.
- Reporte de todos los docentes.
- Reporte de docentes por apellido.

3.1.1.1.3.1.3. Gestión de Alumnos

Los procesos que el sistema podrá realizar son los siguientes:

- Ingreso de nuevos alumnos.
- Paso a inactividad de alumnos.
- Modificación de datos del alumno.

- Reporte de todos los alumnos.
- Reporte de alumnos por ciclo.
- Reporte de alumnos por apellido.

3.1.1.1.3.1.4. Gestión de Notas

Los procesos que el sistema podrá realizar son los siguientes:

- Ingreso de notas por cada alumno matriculado .
- Paso a inactividad de notas, si se deshabilita al estudiante.
- Modificación de notas depende de las fechas topes de cada parcial.
- Reporte individual de las notas de cada alumno.
- Reporte general de notas.

3.1.1.1.3.1.5. Gestión de Faltas

Los procesos que el sistema podrá realizar son los siguientes:

- Ingreso de faltas.
- Modificación de faltas.
- Reporte de faltas de cada estudiante.

3.1.1.1.3.1.6. Gestión de Historial

Los procesos que el sistema podrá realizar son los siguientes:

- Generación automática del historial.
- Reporte de alumnos aprobados.

- Reporte de alumnos reprobados.
- Reporte de cierre de ciclo.

3.1.1.1.3.1.7. Gestión de Cierre del Sistema

Los procesos que el sistema podrá realizar son los siguientes:

- Ingreso de fechas topes para evaluaciones.
- Modificación de fechas topes de evaluaciones.

3.1.1.1.3.1.8. Gestión de Egresados

Los procesos que el sistema podrá realizar son los siguientes:

- Generación automática de alumnos egresados.
- Reporte de egresados.

3.1.1.1.3.2. Requisitos de Rendimiento

El rendimiento del software será óptimo, debido que será desarrollado en C# .NET y SQL Server 2000, son herramientas de última generación para realizar aplicaciones que necesiten soportar grandes flujos de información, así como soporte para cliente servidor en un entorno Intranet.

3.1.1.1.3.3. Requisitos Tecnológicos

3.1.1.1.3.3.1. Requisitos de Hardware

Para el sistema “EBCAL UTC” se recomienda el siguiente hardware para su funcionamiento:

- La máquina Servidor con las siguientes características:
 - Procesador: Pentium IV de 1.4 Ghz. o superior.
 - Memoria RAM: 256 Mb. o superior.
 - Espacio Disco: 40 Gb. o superior.
 - Tarjeta de red: 10/100 o superior.

- Computadoras (clientes) con las siguientes características:
 - Procesador: Pentium III de 866 Mhz. o superior.
 - Memoria RAM: 128 Mb. o superior.
 - Espacio Disco: 30 Gb. o superior.
 - Tarjeta de red: 10/100 o superior.

- Concentrador (Swich o Hub) de 8 puertos o superior.

3.1.1.1.3.3.2. Requisitos de Software

- Sistema Operativo Windows XP Profesional; 2000 Server, 2003 Server.(se recomienda Win XP profesional)
- Sistema Operativo Windows 95 o superior (Clientes).
- Visual Studio .NET 2003(Framework 1.1 para el servidor)
- SQL Server 2000 (para el servidor).
- Internet Information Service 5.5(IIS) (en el servidor)
- Internet Explorer 6.0 (Servidor)
- Internet Explorer 4.0 o superior (Clientes)

3.1.1.1.3.4. Requisitos de Desarrollo

Como requerimiento para el desarrollo del sistema “EBCAL UTC” se considera conveniente emplear el modelo de fases, debido a las características que presenta el software.

3.1.1.1.3.5. Requisitos de Comunicación

Se podrá utilizar la red de datos existente en la Secretaría de la Especialidad; al existir dos equipos de cómputo en la Especialidad de Educación Semipresencial del Centro Asociado Latacunga, se optará por convertir uno de ellos en servidor para la aplicación.

3.1.1.1.3.6. Atributos

3.1.1.1.3.6.1. Seguridad

La seguridad empleada para el sistema “EBCAL UTC” es la siguiente:

- Verificación de ingreso de número de cédula, por medio de algoritmo de autenticación.
- Autorización a las páginas webs para el ingreso a opciones de altas, deshabilitación y cambios; esto se lo hace mediante el logeo de una cuenta de usuario Administrador con su respectiva contraseña que está encriptada en la base de datos con el algoritmo de hash
- Activación de Cookies mediante un determinado tiempo para el cierre del sistema, esto ayuda en caso de inactividad del sistema se bloqueará en forma automática.

3.1.1.1.3.6.2. Tipos de Usuarios

3.1.1.1.3.6.2.1. Súper Administrador del Sistema

Persona encargada del mantenimiento y control general del sistema (altas, deshabilitación y cambios), además este puede modificar notas y faltas cuando el sistema se ha cerrado.

3.1.1.1.3.6.2.2. Secretarios (Administrador)

Gestiona las actividades académicas y está en relación directa con el sistema (altas, deshabilitación, cambios).

3.1.1.1.3.6.2.3. Docentes y Alumnos

Personas pueden consultar información referente al rendimiento académico, historial, etc.

3.1.1.1.3.7 Análisis de requisitos de la base de Datos.

En esta parte se analiza los requerimientos básicos que tiene la Base de Datos mediante la utilización de diagramas de casos de uso y diagramas de secuencias.

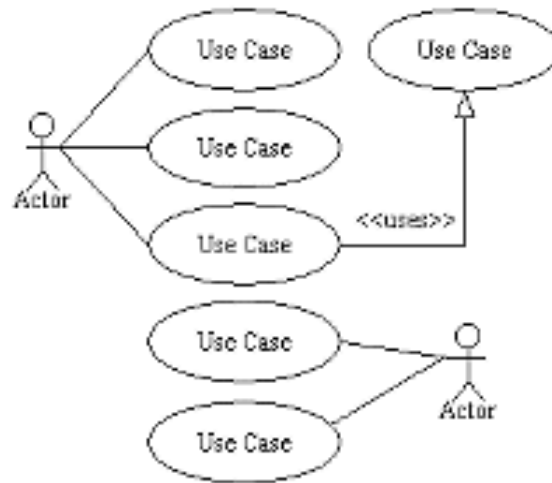
3.1.1.1.3.7.1 Diagrama de Casos de Usos.

Se emplean para visualizar el comportamiento del sistema, una parte de él o de una sola clase. De forma que se pueda conocer como responde esa parte del sistema. El diagrama de uso es muy útil para definir como debería ser el comportamiento de una parte del sistema, ya que solo especifica como deben comportarse y no como están implementadas las partes que define. Por ello es un buen sistema de documentar partes del código que deban ser reutilizables por otros desarrolladores. El diagrama también puede ser utilizado para que los expertos de dominio se comuniquen con los informáticos sin llegar a niveles de complejidad. Un caso de uso especifica un requerimiento funcional, es decir indica esta parte debe hacer esto cuando pase esto.

En el diagrama se presenta con diferentes figuras que pueden mantener diversas relaciones entre ellas:

- ❑ Casos de uso: representado por una elipse, cada caso de uso contiene un nombre, que indique su funcionalidad. Los casos de uso pueden tener relaciones con otros caso de uso. Sus relaciones son:
- ❑ Include: Representado por una flecha, en el diagrama de ejemplo podemos ver como un caso de uso, el de totalizar el coste incluye a dos casos de uso.
- ❑ Extends: Una relación de una caso de Uso A hacia un caso de uso B indica que el caso de uso B implementa la funcionalidad del caso de uso A.
- ❑ Generalization: Es la típica relación de herencia.
- ❑ Actores: se representan por un muñeco. Sus relaciones son:
- ❑ Communicates: Comunica un actor con un caso de uso, o con otro actor.
- ❑ Parte del sistema (System boundary): Representado por un cuadro, identifica las diferentes partes del sistema y contiene los casos de uso que la forman.

FIGURA 6 Diagrama de Casos.



Fuente: <http://grial.uc3m.es/~ffernand/publicaciones/>

Ver Anexo 3 (Diagrama de Casos de Usos, sistema EBCAL UTC)

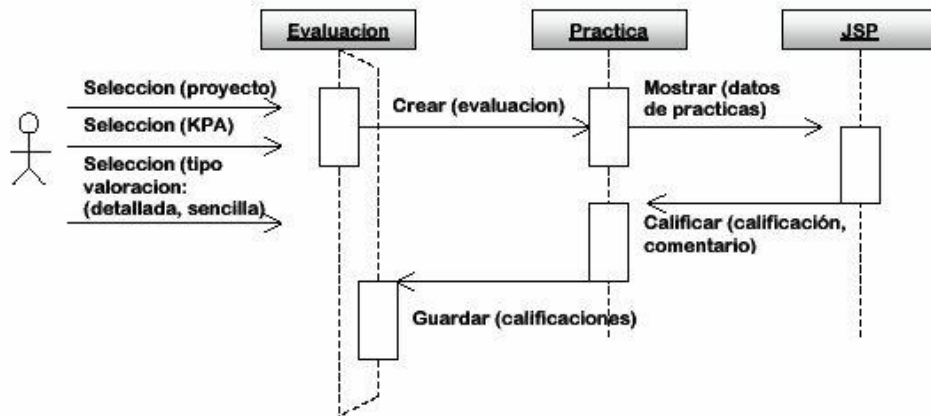
3.1.1.1.3.7.2 Diagrama de Secuencias.

El Diagrama de Secuencia es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema. Un diagrama de secuencia se modela para cada caso de uso. Mientras que el diagrama de caso de uso permite el modelado de una vista 'business' del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes pasados entre los objetos.

Típicamente uno examina la descripción de un caso de uso para determinar qué objetos son necesarios para la implementación del escenario. Si tienes modelada la descripción de cada caso de uso como una secuencia de varios pasos, entonces puedes "caminar sobre" esos pasos para descubrir qué objetos son necesarios para que se puedan seguir los pasos.

¹²Un diagrama de secuencia muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como vectores horizontales. Los mensajes se dibujan cronológicamente desde la parte superior del diagrama a la parte inferior; la distribución horizontal de los objetos es arbitraria. Ver anexo 3 (Diagrama de Secuencias)

FIGURA 7. Ejemplo de Diagrama de Secuencias.



Fuente: http://www.dc.uba.ar/people/materias/isoft1/2001_2/apuntes/Diseno.pdf

3.1.1.2. Diseño

En la fase de diseño se establece el comportamiento dinámico del sistema, es decir, como debe reaccionar ante los acontecimientos. Para su desarrollo también se utilizan los lenguajes de modelado descritos en la fase de especificación. El resultado obtenido de la etapa de diseño facilita enormemente la implementación posterior del sistema, pues proporciona la estructura básica del sistema y como los diferentes componentes actúan y se relacionan entre ellos.

¹² UTC, Internet; <http://grial.uc3m.es/~ffernand/publicaciones/>; Último acceso: miércoles 07 de enero 2004

Una buena metodología de diseño consiste en utilizar patrones. Un patrón de diseño ofrece una solución concreta a los posibles problemas que podamos encontrar en la construcción del sistema, indicando la mejor forma de modelar los objetos. Una vez aplicados, su implementación en lenguajes orientados a objetos no presenta ningún problema adicional.

3.1.1.2.1 Diagrama de Clases.

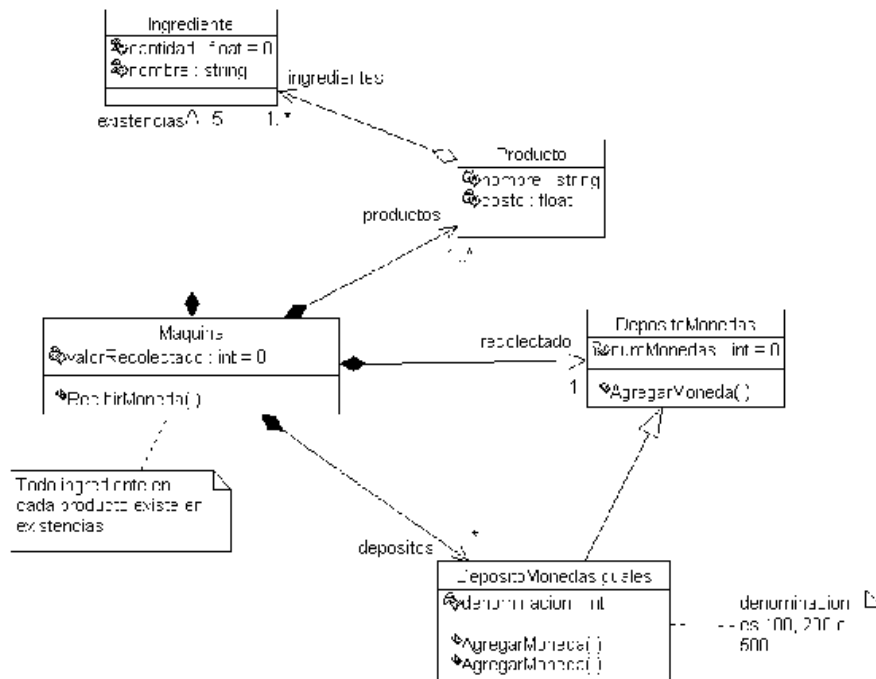
Diagrama de clases (estructura estática)

¹³Los diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y sus interrelaciones (incluyendo herencia, agregación, asociación, etc). Los diagramas de clase son el pilar básico del modelado con UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer (análisis), como para mostrar cómo puede ser construido (diseño). El diagrama de clases de más alto nivel (*main class diagram*), será lógicamente un dibujo de los paquetes que componen el sistema. A su vez cada paquete tendrá un *main class diagram* que muestra las clases del paquete

Las clases se documentan con una descripción de lo que hacen, sus métodos y sus atributos. Las relaciones entre clases se documentan con una descripción de su propósito, su cardinalidad (cuantos objetos intervienen en la relación) y su opcionalidad (cuando un objeto es opcional el que intervenga en una relación). La descripción de clases complejas se puede documentar con diagramas de estados. Ver anexo 3 (Diagrama de Clases)

¹³ UTC, Internet; http://www.dc.uba.ar/people/materias/isoft1/2001_2/apuntes/Diseno.pdf ; Último acceso: martes 13 de enero 2004

Figura 8. Ejemplo de Diagrama de clases.



Fuente: <http://www-gris.det.uvigo.es/~avilas/UML/node37.html>

3.1.1.2.2 Diseño Lógico

Este diseño toma la información brindada por el Diseño Conceptual y la aplica al conocimiento técnico. Mientras que los requerimientos y necesidades de los clientes y usuarios son identificados en la perspectiva de diseño previa, es en éste diseño que la estructura y comunicación de los elementos de la solución son establecidos. Los objetos y servicios, la interfaz de usuario y la base de datos lógica son el conjunto de elementos identificados y diseñados en esta perspectiva.

En esta etapa no interesan los detalles de implementación física, tales como donde se van a alojar ciertos componentes o cuantos servidores están involucrados. El único interés es crear un modelo de abstracción de alto nivel, independiente de cualquier modelo físico.

Este alto nivel de abstracción permite distanciarse de muchos detalles recolectados en la fase conceptual y organizarlos sin tener que analizar los detalles particulares de cada uno de los requerimientos. Además hace posible centrarse en un requerimiento específico a la vez sin perder la visión de la aplicación como un todo.

El Diseño Lógico es el proceso de tomar los requerimientos de usuario obtenidos en el Diseño Conceptual y mapearlos a sus respectivos objetos de negocios y servicios.

Objetivos del diseño lógico

- El objetivo principal es transformar el esquema conceptual de datos en el esquema lógico de datos
- Otros objetivos del diseño lógico son:
 - Eliminar redundancias
 - Conseguir máxima simplicidad
 - Evitar cargas suplementarias de programación para conseguir;
 - una estructura lógica adecuada.
 - un equilibrio entre los requisitos de usuario y la eficiencia.

3.1.1.2.3 Diseño Físico

Es donde los requerimientos del diseño conceptual y lógico son puestos en una forma tangible. Es en este diseño que las restricciones de la tecnología son aplicadas al Diseño

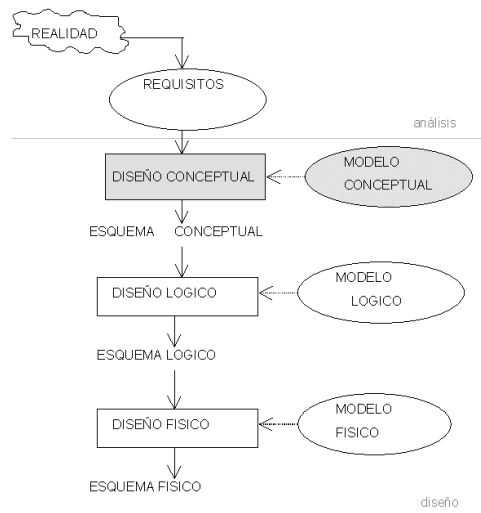
Lógico de la solución. El Diseño Físico define cómo los componentes de la solución, así como la interfaz de usuario y la base de datos física trabajan juntos. Desempeño, implementación, ancho de banda, escalabilidad, adaptabilidad y mantenibilidad son todos resueltos e implementados a través del Diseño Físico. Ya que esta perspectiva transforma los diseños previos en una forma concreta, es posible estimar qué recursos, costos o programación de tiempo serán necesarios para concretar el proyecto.

Al lidiar con estas tres perspectivas, es importante notar que éstas no son series de pasos con puntos de finalización claros. No es necesario alcanzar un punto específico en una de las perspectivas antes de continuar con la siguiente. De hecho, un área de diseño puede ser usada en combinación con otra de manera tal que mientras una parte de la solución es diseñada conceptual o lógicamente, otra esta siendo codificada o implementada en el producto final. Desde que no existen etapas con puntos definidos o límites, es posible regresar a las distintas perspectivas de diseño cuantas veces sea necesario. Esto permite afinar el diseño revisando y rediseñando la solución. Ver anexo 3 (Modelo Físico)

Objetivos del diseño físico

- Minimizar el tiempo de respuesta
- Tiempo entre la introducción de una transacción T de BD y la obtención de respuesta
- Maximizar la productividad de las transacciones
- Optimizar el aprovechamiento del espacio.

FIGURA 9. Ejemplo de diseño Físico



Fuente:<http://www.monografias.com/trabajos14/tecnolcomp/tecnolcomp2.shtml#LOGICO>

3.1.1.2.4 Especificación de las tablas de la Base de Datos .

¹⁴En esta fase se presenta las características específicas de las tablas de la Base de Datos, con sus respectivos campos, tipo de datos y las acciones que se puede ejecutar en cada transacción que el usuario realizará por medio de la interacción con el Sistema EBCAL.

Descripción:

Se construyen sobre cuatro relaciones de componentes. Se pueden utilizar las siguientes combinaciones ya sea individualmente o en conjunción con alguna otra.

Relación secuencial: define los componentes que siempre se incluyen en una estructura de datos.

¹⁴ UTC, Internet; <http://www.monografias.com/trabajos14/tecnolcomp/tecnolcomp2.shtml#LOGICO>; Último acceso: sábado 24 enero 2004

Relación de selección: (uno u otro), define las alternativas para datos o estructuras de datos incluidos en una estructura de datos.

Relación de iteración: (repetitiva), define la repetición de un componente.

Relación opcional: los datos pueden o no estar incluidos, o sea, una o ninguna iteración.

Notación

Los analistas usan símbolos especiales con la finalidad de no usar demasiada cantidad de texto para la descripción de las relaciones entre datos y mostrar con claridad las relaciones estructurales. En algunos casos se emplean términos diferentes para describir la misma entidad (alias) estos se representan con un signo igual (=) que vincula los datos.

3.1.1.3 Prueba del Sistema:

Se llevaron a cabo dos tipos de pruebas en el sistema. Las primeras pruebas funcionales se hicieron para estar seguros que el sistema está funcionando como se espera o como fue diseñado. Estas pruebas se llevaron a cabo creando un grupo de casos de prueba con datos de prueba. Las pruebas incluyeron la validación de campos, reglas de negocio, integración de procesos y la entrada de datos. Las pruebas fueron realizadas en ambos ambientes presentados tanto en el cliente como en el servidor.

Las segundas pruebas fueron realizadas mediante la generación de reportes utilizando como parámetros las notas de alumnos para cada uno de los ciclos pertenecientes a la especialidad de Educación Semipresencial Centro Asociado Latacunga.

Seguidamente se detalla las pruebas realizadas en el sistema “EBCAL”:

En este sistema existe dos formas de ingreso la una es pública; esto quiere decir que tienen acceso todos los estudiantes para consulta de notas y descarga de módulos, y la otra forma para el ingreso del administrador del sistema; este debe tener una contraseña para la utilización del resto del sistema.

3.1.1.3.1 Prueba de Datos:

La prueba de datos consiste en verificar si el usuario ingresó correctamente los campos requeridos para el almacenamiento en la base de datos; como se muestra en la tabla 16; (ver anexo 5 Manual de Usuario).

Tabla 16. Prueba de Datos

ACCIÓN	DATO DE PRUEBA	RESULTADO
Comprobar cédula del alumno	0502521032	Asignaturas con sus notas y profesores respectivos.
Cambio de clave del administrador	Clave anterior + clave nueva + rescribir clave nueva	Se ha cambiado la clave exitosamente
Nomina de estudiantes por ciclo	ciclo	Nomina de estudiantes correspondiente al ciclo solicitado
Nomina de profesores por ciclo	ciclo	Nomina de profesores correspondiente al ciclo solicitado

Fuente: Los Autores.

3.1.1.3.2 Prueba de Validaciones:

La prueba de validaciones consiste en verificar si los datos ingresados cumplen diferentes normativas utilizando diferentes algoritmos; como es el caso de la verificación de la cédula de identidad del estudiante, ingreso mandatorios de nombres,

apellido, etc. de acuerdo a la estructura de la base de datos. (Ver anexo 5 Manual de Usuario).

Tabla 17. Prueba de Validaciones

ACCIÓN	DATO DE PRUEBA	RESULTADO
Ingreso de cédula de identidad	0502523111	Cédula válida
Ingreso de correo electrónico	mal@satnet	Error: verifique si ingresó correctamente el e-mail
Ingreso de nombre	(campo vacío)	Debe ingresar un nombre
Ingreso de fecha de nacimiento	55/abril/2	Error: verifique el día, y el año de nacimiento el rango esta incorrecto!!
Ingreso de notas	12	Error: el rango permitido es de 0-10 puntos.
Nuevo estudiante	0502136328	El alumno ingresado ya existe en la base de datos

Fuente: Los Autores.

3.1.1.3.3 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 el cliente a la página de inicio de sesión. (Ver anexo 5 Manual de Usuario).

FIGURA 10. Autenticación en ASP.NET (página de inicio)



Fuente: Los Autores

Tabla 18. Prueba de Seguridad (cookie)

ACCIÓN	DATO DE PRUEBA	RESULTADO
Sistema inactivo 15 minutos en cualquier parte del sistema	Tiempo transcurrido	Redireccionamiento a la pagina de inicio

Fuente: Los Autores

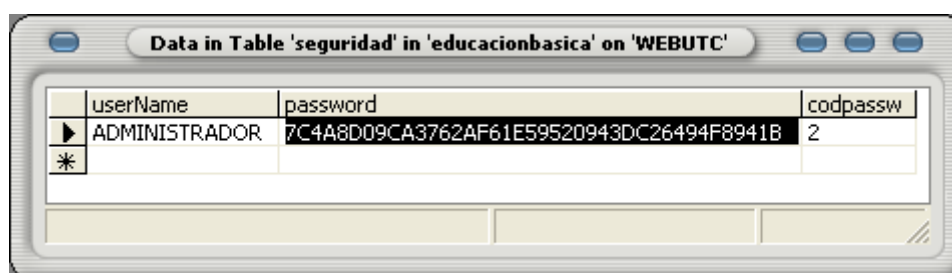
- **Encriptación de claves por medio del algoritmo de HASH**

Para seguridad de la clave de acceso a todo el sistema “EBCAL” se ha visto conveniente utilizar el algoritmo de Hash.

En lugar de almacenar la clave tal cual, se calcula un valor computacionalmente irreversible para cada clave y se almacena dicho valor para confirmar que se ha dado la clave correcta simplemente se aplica el cálculo del punto anterior 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.

“Computacionalmente irreversible” quiere decir que no se puede regresar desde el valor calculado hasta la clave original incluso usando super-computadoras durante largo tiempo. Los algoritmos que hacen este tipo de cálculo son conocidos desde hace rato y se conocen colectivamente como “hashes”. Los dos más populares algoritmos de este tipo tienen el nombre de SHA1 y MD5. (Ver anexo 5 Manual de Usuario).

FIGURA 11. Encriptación Hash (SHA1)



Fuente: Los Autores

Tabla 19. Prueba de Seguridades [Encriptación algoritmo Hash (SHA1)]

ACCIÓN	DATO DE PRUEBA	RESULTADO ENCRIPADO
Ingreso de clave del Administrador	*****	7C4A8D09CA3762AF61E59520943DC26494F8941B

Fuente: Los Autores

3.1.1.4 Implantación del Sistema:

Para la implantación del sistema primero hay que asegurarse que el sistema sea operacional para que los usuarios puedan manipular e interactuar con el sistema EBCAL.

Esta aplicación debe estar en un ambiente de una Intranet ya que es una aplicación cliente servidor; en donde en la máquina servidor estará alojado el motor de Base de datos SQL 2000, así como también estará instalado el sistema EBCAL para lo cual debe estar previamente instalado Framework SDK v1.1 y el Internet Information Service (IIS) como se indica en el anexo 4 Manual del Programador.

Para que todo lo mencionado anteriormente funcione a cabalidad es necesario primordialmente que en la máquina servidor debe estar instalado Windows XP Profesional como sistema operativo.