

ESCUELA POLITECNICA DEL EJÉRCITO

DPTO. DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
ANÁLISIS, DISEÑO, DESARROLLO E IMPLEMENTACIÓN  
DE UN SISTEMA DE CONTROL DE COBRO DE  
MATRÍCULA Y PENSIONES PARA LA UNIDAD EDUCATIVA  
PARTICULAR MARÍA MONTESSORI

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS

*POR: WAGNER EUDIMIO ANDRADE CEDEÑO*

SANGOLQUÍ, martes, 14 de diciembre de 2010

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue realizado en su totalidad por el Sr. WAGNER EUDIMIO ANDRADE CEDEÑO como requerimiento parcial a la obtención del título de INGENIERO EN SISTEMAS E INFORMÁTICA.

Lunes, 28 de julio del 2008

---

Ing. Mauricio Campaña

## **DEDICATORIA**

Todo el esfuerzo y el logro alcanzado no hubiese sido posible sin el apoyo que mi familia me ha dado, por eso dedico todo mi trabajo a mis padres, hermanos y mis bellos hijos, pues ellos son la única fuente de motivación y de superación.

Así mismo dedico de manera especial este trabajo a mi madre querida Teresa Cedeño Pino, quien con todo su esfuerzo siempre me dijo un sigue adelante mijo tu puedes, madre eres el mejor ejemplo a seguir.

**Wagner Eudimio Andrade Cedeño**

## **AGRADECIMIENTO**

Agradezco a mis padres, hermanos, amigos por darme el apoyo día a día para que todas mis metas vayan cumpliéndose.

También agradezco de forma especial a la Unidad Educativa Particular María Montessori por haberme permitido desarrollar este proyecto, el cual ha sido de gran ayuda para el crecimiento interno de la institución.

**Wagner Eudimio Andrade Cedeño**

## INDICE DE CONTENIDO

RESÚMEN.....	9
PRESENTACIÓN .....	11
CAPÍTULO I	
GENERALIDADES.....	12
1.1.- Introducción.....	12
1.2.- Planteamiento del problema.....	14
1.3.- Justificación.....	15
1.4.- Objetivos.....	17
1.4.1. Objetivo general .....	17
1.4.2. Objetivos específicos.....	17
1.5.- Alcance.....	18
1.6.- Factibilidad .....	20
1.6.1. TÉCNICA.....	20
1.6.2. Operativa .....	21
1.6.3. Económica .....	21
CAPITULO II	
MARCO TEORICO .....	24
2.1. Arquitectura windows y web .....	24
Aplicación windows .....	25
Aplicación web .....	27
2.1.1. Visual basic.....	30
2.1.1.1. Nuevas características en visual basic 2008.....	31
2.1.2. C# .....	33
2.1.2.1. Nuevas características en visual c# 2008 .....	34
2.1.3. HTML .....	34
2.1.4. SQL.....	35
2.1.5. SQL server.....	36
2.1.5.1. Nuevas características en SQL server 2008.....	37
2.1.6. IIS (Internet Information Server).....	39
2.2. Términos relacionados .....	41
2.2.1. Servidor web.....	41
2.2.2. Portal web.....	41
2.2.3. Página dinámica .....	42
2.2.4. Web services .....	43
2.2.5. Intranet .....	43
2.3. Justificación de las herramientas utilizadas.....	44
CAPITULO III	
METODOLOGÍA.....	47
Metodología de desarrollo de sistemas .....	47
3.1. Metodología MSF (Microsoft Solution Framework).....	50
Componentes del MSF .....	51

<i>Principios</i> .....	51
Disciplinas .....	51
Gestión de proyectos.....	51
Control de riesgos .....	52
Control de cambios .....	52
Modelos.....	52
Equipo de trabajo .....	52
Proceso .....	53
3.2. Ejecución de la metodología.....	54
3.2.1. Requisitos del sistema .....	54
3.2.2. Visión (visión y alcance aprobados).....	60
3.2.2.1. Documento visión.....	60
3.2.2.2. Documento detalle de visión .....	64
3.2.2.3. Documento matriz de riesgos.....	77
3.2.2.4. Acta de aprobación de visión .....	78
3.2.3. Planeación (cronograma de proyecto aprobado) .....	79
3.2.4. Desarrollo(alcance completo) .....	80
3.2.5. Estabilización( versión aprobada) .....	81
3.2.6. Instalación(entregas) .....	81
3.2.7. Soporte(entrega ajustada) .....	82
 CAPITULO IV	
CONCLUSIONES Y RECOMENDACIONES .....	83
4.1. Conclusiones.....	83
4.2. Recomendaciones.....	85
Glosario.....	87
Bibliografía .....	88

## Listado de Cuadros

Cuadro 1.1: (Cuadro Referencial de Costos del Proyecto (Empresa)) .....	22
Cuadro 1.2: (Necesidades en Hardware, Software y costos (Empresa)) .....	23
Cuadro 2.1: (Lenguajes de aplicación utilizado en cada capa para la aplicación Windows) .....	29
Cuadro 2.2: (Lenguajes de aplicación utilizado en cada capa para la aplicación Web)....	29
Cuadro 2.3: (Comparativo a nivel del Frontend) .....	45
Cuadro 2.4: (Comparativo a nivel del Backend).....	46
Cuadro 3.1: (Modelo de caso de uso real).....	66
Cuadro 3.2: (Plan de instalación del sistema) .....	73
Cuadro 3.3: ( Visión y Alcance Aprobados) .....	79

## Listado de Figuras

<b>Figura 2.1:</b> (Arquitectura Windows por capas) .....	25
<b>Figura 2.2:</b> (Diseño de Tablas que esta en constante crecimiento) .....	26
<b>Figura 2.3:</b> (Diseño de Tablas Administrativas) .....	27
<b>Figura 2.4:</b> (Arquitectura Web por capas y componentes).....	28
<b>Figura 3.1:</b> (Modelo del Equipo de Trabajo MSF) .....	53
<b>Figura 3.2:</b> (Modelo de Proceso de MSF).....	54
<b>Figura 3.3:</b> (Diseño propuesto de interfaz Windows).....	61
<b>Figura 3.4:</b> (Diseño de interfaz Windows real) .....	62
<b>Figura 3.5:</b> (Diseño de interfaz Web) .....	63
<b>Figura 3.6:</b> (Diseño de interfaz Web real) .....	64
<b>Figura 3.7:</b> (Modelo caso de uso).....	70
<b>Figura 3.8:</b> (Diagrama de Componentes) .....	75
<b>Figura 3.9:</b> (Diagrama de Plataforma o despliegue) .....	76
<b>Figura 3.10:</b> (Caso de uso general).....	77
<b>Figura 3.11:</b> (Riesgos y acciones a tomar) .....	78
<b>Figura 3.12:</b> (Diseño de tiempos de la visión y alcance aprobados).....	80



## RESÚMEN

La Unidad Educativa Particular María Montessori requiere de una aplicación Windows y Web que permita contar con un sistema de Matriculación y cobro de Pensiones que cumpla con requerimientos tecnológicos que mejoren su nivel competitivo académico.

El proyecto contempla cuatro capítulos en los que se han considerado parámetros que han permitido el correcto desarrollo de las aplicaciones Windows y Web.

Capítulo I. Se detallan aspectos que permiten tener una visión general del proyecto, como son: introducción, planteamiento del problema, justificación, objetivos, alcance, factibilidad técnica y económica.

Capítulo II. Se realiza una descripción de la arquitectura Windows y Web tres capas, herramientas de desarrollo empleadas en cada capa, características de herramientas de apoyo y terminología relacionada.

Capítulo III. Aquí se mencionan aspectos metodológicos aplicados en el desarrollo, en los que consta, la metodología RUP con una descripción de sus fases y un análisis del lenguaje de modelamiento UML y los diseños que incluye el modelado.

Capítulo IV. Al término del proyecto se presentan las conclusiones y recomendaciones, establecidas del análisis y desarrollo del Sistema Académico orientado a Windows y WEB usando WEB SERVICES para la Unidad Educativa Particular María Montessori.

## PRESENTACIÓN

El continuo desarrollo de la tecnología, ha llevado a los centros educativos del Ecuador a contar con sistemas informáticos que permitan mejorar la gestión de los procesos, garanticen la solidez de la información y en general, proporcionen nuevas formas de administración, gerencia y control en los pagos de matriculas y pensiones en los centros educativos.

En otras palabras, la disponibilidad de un sistema de información avanzado implica la adopción de un cambio en los procesos tradicionales y permite poner la información relevante a disposición de la empresa en línea sea esta a nivel Windows o desde el Internet.

Nuestro objeto de estudio se centra en el control de cobros de matriculas y pensiones para la Unidad Educativa Particular María Montessori, dicha información, en la actualidad se maneja de forma manual, limitando la toma de decisiones y condiciones reales de pagos actualizados.

La necesidad de contar con aplicaciones que permitan conocer la realidad actual en el centro educativo, es evidente e indispensable debido a que del control de estos valores depende el funcionamiento de todas sus actividades.

# CAPÍTULO I

## GENERALIDADES

### 1.1.- Introducción

La Unidad Educativa Particular María Montessori, ha tenido y tiene la necesidad de contar con sistemas que permitan automatizar sus procesos internos. Una de estas necesidades es el cobro de Matrícula y Pensiones a todos los estudiantes.

Automatizar estos dos procesos permitirá mejorar la velocidad de respuesta a reportes diarios del cobro de pensiones de forma real, ordenada y en línea.

Actualmente el cobro de pensiones es realizado en forma manual, lo que causa demora y resultados no completos como son:

Para la directora:

- Lista de estudiantes que adeudan pensiones para fechas establecidas.
- Establecer políticas internas como son: Acuerdo de pagos y manejo de becas.
- Detalle en el manejo contable de ingresos por matriculas y pensiones.

Para la secretaria:

- El cobro de pensiones en días con mayor demanda como fin de mes en los que por el volumen de cobros fácilmente se puede cometer errores.
- Demora al cobro debido a las diferentes búsqueda manuales que tiene que realizarse para identificar al estudiante.
- Demasiado tiempo para presentar reportes.

Para los docentes:

- Imprimir el reporte de estudiantes al día con los pagos

## 1.2.- Planteamiento del Problema

Actualmente la Unidad Educativa Particular María Montessori realiza un manejo básico y limitado de su información financiera, realizando en forma manual su manejo contable.

Cuando la información de cualquier institución que cada día esta creciendo se la maneja manualmente, impide la toma de decisiones, el crecimiento de la institución, da paso a redundar mucho en la información y cometer errores contables, es imposible tener reportes importantes y actualizados en tiempos cortos; bajo este contexto se presentan los siguientes problemas.

- Archivos Planos
- Inseguridad en la información (duplicidad)
- Resultados no apegados a la realidad
- Pérdida de información
- Limitación para la realización de tareas
- No presenta facilidades para el manejo de la información (Reportes en línea)

Considerando las prioridades antes mencionadas, la Unidad Educativa Particular María Montessori se ha visto en la necesidad obligada de automatizar sus procesos de Matrícula y cobro de Pensiones desarrollando un sistema informático que cumpla cierto tipo de características de acuerdo a los requerimientos internos, administrativos, financieros, y de seguridad.

### **1.3.- Justificación**

Automatizar los procesos cobros de matriculas y pensiones, utilizando un portal Web principalmente para reportes, permitirá que la información sea difundida y accesible donde y cuando sea requerida; es decir que la información puede ser consultada en cualquier momento y desde cualquier computadora en la intranet de la unidad educativa.

Mediante la implementación de estas dos tecnologías Windows y Web se puede brindar una mejor atención a los padres de familia de esta institución, entregando no solo información sino también servicios que beneficien el desempeño de las actividades y el trabajo académico.

Además de contar con un mejor servicio, la imagen de la Unidad Educativa Particular María Montessori incursionaría en un campo más amplio como es el de la Internet, permitiendo que la institución sea difundida no solo a nivel local sino también a nivel mundial debido al uso masivo que esta tecnología tiene actualmente.

Dentro de las características principales a nivel Windows se tienen:

- Cobro de Matricula
- Cobro de Pensiones
- Reportes detallados de Matriculas cobradas y por cobrar
- Reportes detallados de Pensiones cobradas y por cobrar

Dentro de las características principales del Portal destacan:

- Información Organizacional de la Unidad Educativa Particular María Montessori
- Cobro de Matricula
- Reportes detallados de Matriculas cobradas y por cobrar
- Reportes detallados de Pensiones cobradas y por cobrar



## **1.4.- Objetivos**

### **1.4.1. Objetivo general**

- Realizar el análisis, diseño, implementación y pruebas del cobro de matrícula y pensiones a través de la Intranet de la Unidad Educativa Particular María Montessori por medio de un browser.

### **1.4.2. Objetivos específicos**

- Describir el proceso a seguir para poder realizar la matrícula y el cobro de pensiones a los estudiantes.
- Utilizar arquitectura web para cargar el sistema desde cualquier punto de la red de UEPMM<sup>1</sup>.
- Desarrollar el sistema bajo una arquitectura cliente servidor multicapas a nivel Windows y Web.
- Aplicar como metodología MSF (Microsoft Solution Framework) y la notación UML (Leguaje de Modelamiento Unificado) para el modelamiento.
- Realizar el análisis de los procesos de matrícula y pensiones en la UEPMM.
- Analizar las distintas arquitecturas donde se puede crear Web dinámicas.

## 1.5.- Alcance

El sistema para cobro de matrícula y pensiones, está basado y enfocado a presentar respuesta actualizada de pagos de matrículas y pensiones. El producto final será un sistema distribuido multicapas orientado a Windows y Web, que permitirá al personal administrativo y profesores obtener reportes actualizados sobre el cobro de matriculas y pensiones.

Ambos ambientes estarán conectados a una misma base de datos que permitirá a nivel Windows manejar lo cobros correspondientes y reportes necesarios, desde la intranet el manejo de reportes requeridos por los docentes.

En el módulo de administración se permitirá:

- Crear los usuarios autorizados para:
  - Crear, modificar, estudiantes en la UEPMM.
  - Imprimir reportes de pagos de matriculas, pensiones, estudiantes actuales.
  - Realizar los cobros de matrícula y pensiones.

El docente tendrá una interfaz intuitiva y fácil de usar con un entorno Web, la que le permitirá únicamente general reportes de pensiones cobradas y por cobrar.

La secretaría podrá matricular, cobrar pensiones e imprimir reportes de los cobros realizados y pendientes. El ambiente que manejará será Windows y tendrá

limitaciones para eliminar pensiones cobradas en caso de ser necesario. Tiene acceso a imprimir reportes utilizando el entorno Windows y Web.

El administrador podrá crear, modificar, eliminar estudiantes y configurar decisiones de cobros de matrículas y pensiones para cada año lectivo. El ambiente que manejará será Windows y tiene la capacidad para eliminar pensiones cobradas. Tiene acceso a imprimir reportes utilizando los dos ambientes Windows y Web.

La Documentación que se elaborará dentro del proyecto es la siguiente.

- Manual del Sistema
- Manual de Usuario

## **1.6.- Factibilidad**

### **1.6.1. Técnica**

#### Información del tema

Se cuenta con manuales en la UEPMM que permite de los procesos a seguir para realizar los cobros de matrículas y pensiones.

#### Asesoramiento técnico

La directora de la UEPMM nos dio la apertura a reuniones con su personal administrativo que conoce perfectamente del proceso de matrícula y pensiones. Además se tiene la colaboración del Ing. Mauricio Campaña, que cuenta con bastos conocimientos en el desarrollo de aplicaciones Windows y Web.

#### Plataforma tecnológica

El proyecto requiere de un equipo de hardware dedicado para el funcionamiento específico de la aplicación, el cual será proporcionado por la UEPMM.

### **1.6.2. Operativa**

La UEPMM a través de su directora está dispuesta a proporcionar el apoyo necesario para obtener los requerimientos e implantar el sistema. No existirá resistencia al cambio debido a que esta solución resolverá un problema que tiene la institución (Véase la carta de auspicio).

### **1.6.3. Económica**

El siguiente cuadro detalla los gastos necesarios para el desarrollo del proyecto, tomando en cuenta diferentes aspectos como bibliografía, materiales de oficina, depreciación de equipo, diseñador, programadores.

Cuadro 1.1: (Cuadro Referencial de Costos del Proyecto (Empresa))

<b>Concepto</b>	<b>Costo/Semana (\$)</b>	<b>Semanas</b>	<b>Costo Total (\$)</b>
-----------------	------------------------------	----------------	-----------------------------

<b>Ingresos</b>			
Aporte Tesista			5650
<b>Total Ingresos</b>			<b>5.650</b>

<b>Egresos</b>			
Personal			
Diseñador			300
Programador	160	20	3.200
Recursos			
Mat. Bibliográfico			200
Internet	15	20	300
Mat. Oficina/Copias			500
Uso de equipo			350
Derechos de grado			800
<b>Total Egresos</b>			<b>5.650</b>

Cuadro 1.2: (Necesidades en Hardware, Software y costos (Empresa))

<b>Concepto</b>	<b>Costo/Semana (\$)</b>	<b>Semanas</b>	<b>Costo Total (\$)</b>
<b>Hardware</b>			
Computador HP			1.500
<b>Software</b>			
SQL Server 2008 32bit ( Para un Usuario)			500
Visual Studio 2008			2.000
<b>Total</b>			<b>4.000</b>

En conclusión el proyecto tiene un costo **TOTAL: \$9.650**

La UEPMM facilitará las licencias del software así como el hardware para el desarrollo del proyecto.

## CAPITULO II

### MARCO TEORICO

#### 2.1. Arquitectura Windows y Web

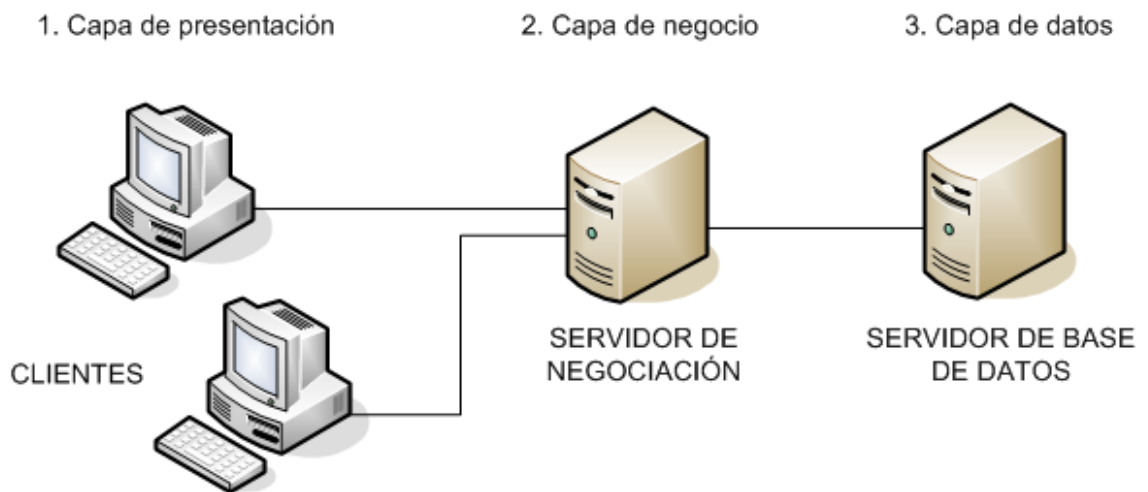
La arquitectura de una aplicación es la vista conceptual de la estructura de esta. Toda aplicación contiene código de presentación, código de procesamiento de datos y código de almacenamiento de datos. La arquitectura de las aplicaciones difiere según como está distribuido este código.

La arquitectura de las aplicaciones Windows y Web se desarrolla en un esquema de tres capas.

La primera capa de presentación (interfaz) sea esta Windows o Web, a nivel Web el servidor Web( Internet Information Server) que es el responsable de dar a los datos un formato adecuado. La segunda capa está referida habitualmente a algún tipo de programa o *script* también se conoce como reglas del negocio, en esta capa se puede tener componentes que permitan el funcionamiento de la interfaz y que puedan interactuar con los datos. Finalmente, la capa tres proporciona a la segunda los datos necesarios para su ejecución.

Una aplicación Windows ó Web típica recogerán datos del usuario (Capa de presentación), los enviará al servidor, que ejecutará un programa (Capa de negocio y datos) y cuyo resultado será formateado y presentado al usuario (Capa de presentación otra vez).





**Figura 2.1:** (Arquitectura Windows por capas)

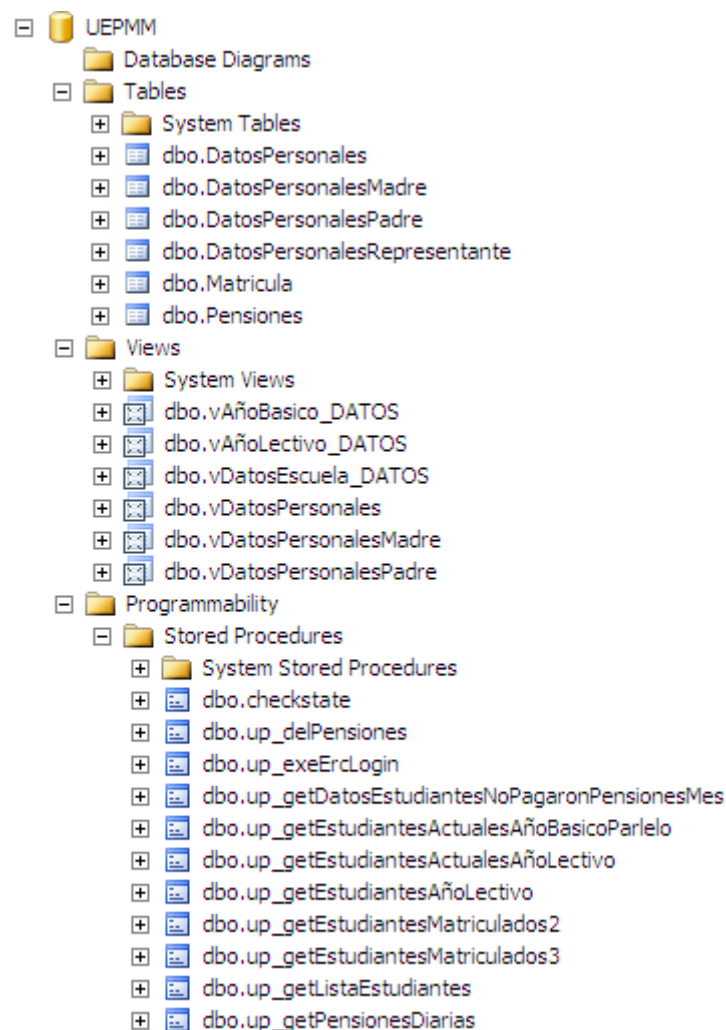
## Aplicación Windows

Capa de Presentación: El **Frontend** ó interfaz está codificada en Visual Basic 2008, utilizando la programación orientada a objetos, y el uso de componentes se brinda la seguridad necesaria para realizar el ingreso bajo un algoritmo de encriptación la petición de información a las reglas del negocio.

Capa de Negocio: Está codificada en Visual Basic 2008 y SQL, esta maneja casi en su totalidad el uso de Store Procedure, que se comunican directamente al Backend ó Base de Datos que en nuestro caso es SQL Server 2008.

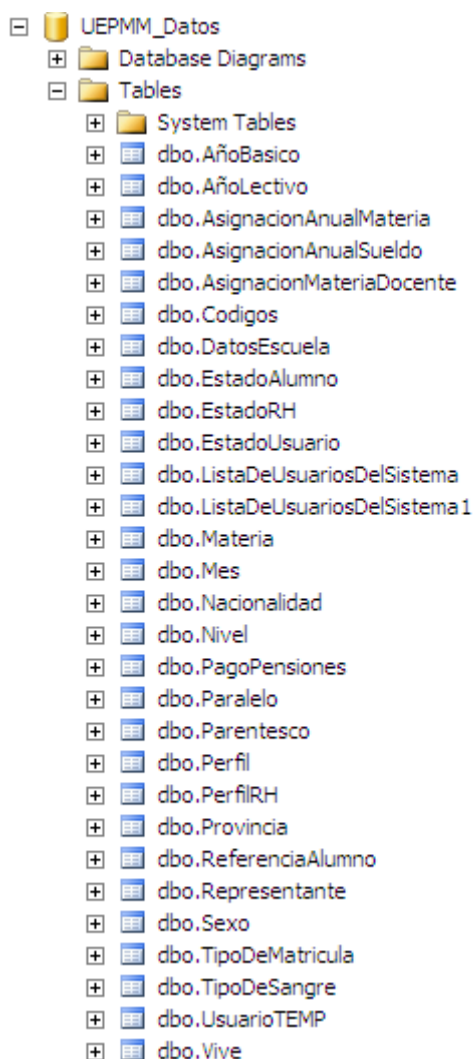
Capa de datos: Como Backend único utilizado tenemos el SQL Server 2008, donde encontraremos el diseños de dos Base de Datos, la primera denominada UEPMM y la segunda UEPMM\_Datos las que manejan por separada la siguiente información.

**UEPMM:** Diseño de las tablas que están en constante crecimiento, las vistas y store procedure que interactúan directamente con la capa de negocios.



**Figura 2.2:** (Diseño de Tablas que esta en constante crecimiento)

**UEPMM\_Datos:** Diseño de las tablas que manejan la parte administrativa, y políticas planteadas en la Unidad Educativa Particular María Montessori.



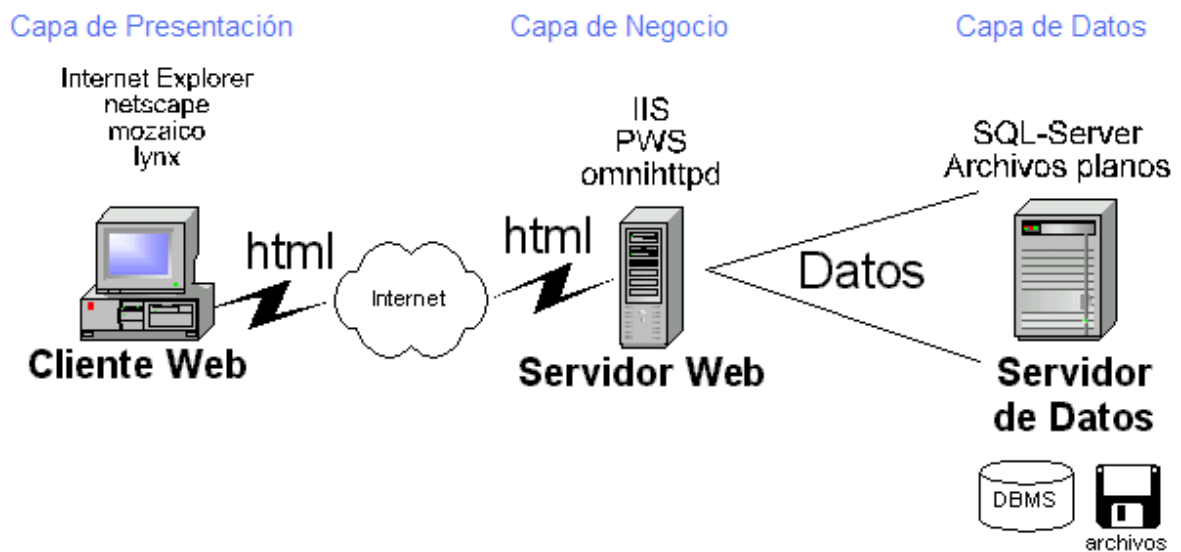
**Figura 2.3:** (Diseño de Tablas Administrativas)

## Aplicación Web

Capa de Presentación: El **Frontend** está codificado en ASP.NET utilizando el lenguaje HTML.

Capa de Negocio: Está codificada en C# 2008 y SQL, esta maneja casi en su totalidad el uso de clases que realizan llamadas a Store Procedure, que se comunican directamente al Backend ó Base de Datos que en nuestro caso es SQL Server 2008.

Capa de datos: Se maneja el mismo diseño que en la aplicación Windows (capa de datos), debido a que el sistema maneja dos ambientes Windows y Web pero que interactúan a la misma base de datos.



**Figura 2.4:** (Arquitectura Web por capas y componentes)

En el siguiente cuadro se describe el lenguaje de aplicación utilizado en cada capa dentro de la arquitectura:

Cuadro 2.1: (Lenguajes de aplicación utilizado en cada capa para la aplicación Windows)

<b>Windows</b>		
<b>Interfaz Usuario</b>	<b>Servidor de Componentes</b>	<b>Motor de Base de Datos</b>
Cliente Windows Visual Basic 2008	DCOM ( <i>Distributed Component Object Model</i> ó Modelo de Componente de Objeto Distribuido), COM+	SQL Server 2008 (Tablas, Vistas, Store Procedure)

Cuadro 2.2: (Lenguajes de aplicación utilizado en cada capa para la aplicación Web)

<b>Web</b>		
<b>Interfaz Usuario</b>	<b>Servidor de Componentes</b>	<b>Motor de Base de Datos</b>
Cliente Web HTML	Servidor Web (Internet Information Server) HTML, Visual C# 2008, SQL	SQL Server 2008 (Tablas, Vistas, Store Procedure)

### **2.1.1. Visual Basic**

Visual Basic es un lenguaje de programación desarrollado por Alan Cooper para Microsoft. El lenguaje de programación es un dialecto de BASIC, con importantes añadidos. Su primera versión fue presentada en 1991 con la intención de simplificar la programación utilizando un ambiente de desarrollo completamente gráfico que facilitara la creación de interfaces gráficas y en cierta medida también la programación misma. En 2001 Microsoft propone abandonar el desarrollo basado en la API Win32 y pasar a trabajar sobre un framework o marco común de librerías independiente de la versión del sistema operativo, .NET Framework, a través de Visual Basic .NET (y otros lenguajes como C-Sharp (C#) de fácil transición de código entre ellos) que presenta serias incompatibilidades con el código Visual Basic existente.

Visual Basic constituye un IDE (entorno de desarrollo integrado o en inglés Integrated Development Environment) que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código (programa donde se escribe el código fuente), un depurador (programa que corrige errores en el código fuente para que pueda ser bien compilado), un compilador (programa que traduce el código fuente a lenguaje de máquina), y un constructor de interfaz gráfica o GUI (es una forma de programar en la que no es necesario escribir el código para la parte gráfica del programa, sino que se puede hacer de forma visual).

Algunas características principales:

- Lenguaje de fácil aprendizaje pensado tanto para programadores principiantes como expertos, guiado por eventos, y centrado en un motor de formularios que facilita el rápido desarrollo de aplicaciones gráficas.
- Admite el polimorfismo mediante el uso de las Interfaces.
- No admite la herencia.
- No requiere de manejo de punteros.
- Posee un manejo muy sencillo de cadenas de caracteres.
- Posee varias bibliotecas para manejo de bases de datos, pudiendo conectar con cualquier base de datos a través de ODBC (Informix, DBase, Access, MySQL, SQL Server, PostgreSQL, etc.) a través de ADO.

#### **2.1.1.1. Nuevas características en Visual Basic 2008**

Esta es la versión original de Visual Basic .NET. Fue publicada junto con Visual C# en 2002 como uno de los lenguajes disponibles para la entonces nueva plataforma de programación .NET de Microsoft.

Actualmente existen 4 versiones publicadas:

- Visual Basic .NET (VB 7)
- Visual Basic .NET 2003 (VB 7.1)
- Visual Basic 2005 (VB 8.0)
- Visual Basic 2008 (VB 9.0)

Visual Basic 2008 es la siguiente evolución de Visual Basic .NET. Fue liberado (RTM) el 17 de Noviembre 2007 en inglés, y en castellano el 2 de Febrero de 2008. Incorpora la versión 3.5 del Framework .NET

**Evoluciones previstas Visual Basic 'VBx' (VB 10.0)** Visual Basic 10, también conocido como VBx, ofrecerá soporte para entorno de ejecución dinámico.

Está previsto que VB 10 forme parte de Microsoft Silverlight 1.1.

#### Características del Visual Basic 2008:

- No es compatible hacia atrás con Visual Basic.
- Lenguaje de programación orientado a objetos.
- Los programas escritos en VB.NET requieren el Framework .NET para ejecutarse en el caso de Visual Basic 2008 la versión del Framework que este utilice.
- El IDE de Visual Studio 2008 permite trabajar contra 3 .NET frameworks diferentes:
  - .NET Framework 2.0
  - .NET Framework 3.0
  - .NET Framework 3.5
- Soporte extendido para arquitecturas de 64 bits.
- Soporte para WebServices usando ASP.NET.
- Servidor web local para realizar pruebas sin depender de IIS.
- Integración con SQL Server 2008.



### 2.1.2. C#

C# fue diseñado por Microsoft, y posteriormente estandarizado por el organismo ECMA. C# intenta aprovechar, en la medida de lo posible, las características de la plataforma .NET. Este nuevo lenguaje, evolución natural de Java y C++, también intenta facilitar a los programadores el cambio de plataforma, por su parecida sintaxis al lenguaje Java.

Sus principales creadores son Scott Wiltamuth y Anders Heljlsberg, éste último conocido por haber sido el diseñador del lenguaje Turbo Pascal y la herramienta de diseño rápido Delphi. En la plataforma .NET es posible escribir código en muchos otros lenguajes, pero C# es el único que ha sido diseñado específicamente para ser utilizado en ella. Utilizar este lenguaje para programar en ella es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes que nos provee la plataforma .NET. Es por esta razón que se suele decir que **C# es el lenguaje nativo de .NET.**

Características principales:

- Orientado a objetos
- Orientado a componentes
- Seguridad de tipos de datos
- Compatible para facilitar la migración de programadores de C++ o Java a C#
- No permite el uso de punteros

### **2.1.2.1. Nuevas características en Visual C# 2008**

- Lenguaje y compilador: C# admite ahora tipos genéricos
- Usa una copia local de Servicios de Internet Information Server (IIS).
- Editor de código: Fragmentos de código, refactorización, entorno de desarrollo
- Entorno de desarrollo
- Documentación y especificaciones del lenguaje
- Depuración
- Administración del diseño de las ventanas
- Explorador de etiquetas en el Diseñador de páginas Web

### **2.1.3. HTML**

El HTML no es más que una aplicación del SGML (Standard Generalized Markup Language), un sistema para definir tipos de documentos estructurados y lenguajes de marcas para representar esos mismos documentos. El término HTML se suele referir a ambas cosas, tanto al tipo de documento como al lenguaje de marcas.

El HTML, Hyper Text Markup Language (Lenguaje de marcación de Hipertexto) es el lenguaje de marcas de texto utilizado normalmente en la www (World Wide Web). Fue creado en 1986 por el físico nuclear Tim Berners-Lee; el cual tomó dos herramientas preexistentes: El concepto de Hipertexto (Conocido también como link o ancla) el cual permite conectar dos elementos entre sí y el

SGML (Lenguaje Estándar de Marcación General) el cual sirve para colocar etiquetas o marcas en un texto que indique como debe verse. HTML no es propiamente un lenguaje de programación como C#, Visual Basic, etc., sino un sistema de etiquetas. HTML no presenta ningún compilador, por lo tanto algún error de sintaxis que se presente éste no lo detectará y se visualizara en la forma como éste lo entienda.

El entorno para trabajar HTML es simplemente un procesador de texto, como el que ofrecen los sistemas operativos Windows (Bloc de notas), UNIX (el editor vi o ed) o el que ofrece MS Office (Word). El conjunto de etiquetas que se creen, se deben guardar con la extensión .htm o .html, estos documentos pueden ser mostrados por los visores o "browsers" de paginas Web en Internet, como Netscape Navigator, Mosaic, Opera y Microsoft Internet Explorer.

También existe el HTML Dinámico (DHTML), que es una mejora de Microsoft de la versión 4.0 de HTML que le permite crear efectos especiales como, por ejemplo, texto que vuela desde la página palabra por palabra o efectos de transición al estilo de anuncio publicitario giratorio entre página y página.

#### **2.1.4. SQL**

SQL (Standar Query Lenguaje) es un lenguaje estandarizado de base de datos, el cual nos permite realizar tablas y obtener datos de ella de manera muy sencilla.

SQL es un lenguaje que consta de varias partes

- Lenguaje de definición de datos ( DDL): Proporciona ordenes para definir esquemas de relación, eliminar relaciones, crear índices y modificar esquemas de relación.
- Lenguaje de manipulación de datos interactivos (DML): incluye un lenguaje de consultas que permite rescatar datos de las relaciones. También incluye ordenes para insertar, suprimir y modificar tuplas.
- Lenguaje de manipulación de datos inmerso (DML): La forma inmersa de SQL esta diseñada para usar dentro de los lenguajes de programación de lenguaje general.
- Definición de vistas (DDL): incluye ordenes para definir vistas.

#### **2.1.5. SQL Server**

Base de datos desarrollada por Microsoft. Su primera versión fue El Standard Edition de SQL Server fue lanzado en abril de 1996.

Algunas características de esta base de datos:

- Escalabilidad: Se adapta a las necesidades de la empresa, soportando desde unos pocos usuarios a varios miles. Empresas centralizadas u oficinas distribuidas, replicando cientos de sites.

- **Potencia:** Microsoft SQL Server es la mejor base de datos para Windows NT Server. Posee los mejores registros de los benchmarks independientes (TCP) tanto en transacciones totales como en coste por transacción.
- **Gestión:** Con un completo interfaz gráfico que reduce la complejidad innecesaria de las tareas de administración y gestión de la base de datos.
- **Orientada al desarrollo:** Visual Basic, Visual C++, Visual J++, Visual Interdev, Microfocus Cobol y muchas otras herramientas son compatibles con Microsoft SQL Server.

#### **2.1.5.1. Nuevas características en SQL Server 2008**

- **Tipos de datos.-** Fecha y hora Geográficas, estructuras jerárquicas Ej.

##### Date

- Solo tipo de fecha // 2008-02-01
- Rango 0001-01-01 a 9999-12-31

##### Time

- Solo tipo hora // 11:48:01.1234567

##### DateTimeOffset

- DateTime con manejo de zona horaria
- 2008-02-01 11:48:01 a.m. +03:00

## DateTime2

- DateTime con mayor precisión
  - 2008-02-01 11:48:19.1234567
  
  - Hierarchy Id
  - Representa una posición en un árbol
  
  - Espaciales
  - Manejo de datos geográficos
- Documentos.- FileStream , APIs administración
  - T-SQL.- Mejoras de código, dependencia de objetos
  - Seguridad.- Encriptación, auditoría
  - Mejoras en los esquemas de validación

## Validaciones Lax

- A través de declaraciones any, anyType, anyAttribute

## Soporte xs:dateTime

- Datos fecha/hora

## Union y List

- Permite mezclar una lista de definiciones y restricciones en un mismo tipo
- Mejoras en Xquery Cláusula “let”

### **2.1.6. IIS (Internet Information Server)**

Es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del *Option Pack* para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Este servicio convierte a un ordenador en un servidor de Internet o Intranet es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente (servidor web).

El servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

Versiones:

IIS 1.0, Windows NT 3.51 Service Pack 3

IIS 2.0, Windows NT 4.0

IIS 3.0, Windows NT 4.0 Service Pack 3

IIS 4.0, Windows NT 4.0 Option Pack

IIS 5.0, Windows 2000

IIS 5.1, Windows XP Professional

IIS 6.0, Windows Server 2003 y Windows XP Professional x64 Edition

IIS 7.0, Windows Vista y Windows Server 2008

IIS fue inicialmente lanzado como un conjunto de servicios basados en Internet para Windows NT 3.51. IIS 2.0 siguió agregando soporte para el sistema operativo Windows NT 4.0 e IIS 3.0 introdujo las páginas activas de servidor, esta tecnología es una plataforma de scripting dinámico.

IIS 4.0 eliminó el soporte para el protocolo Gopher y fue puesto con Windows NT como un CD-ROM de "Paquete Opcional" separado.

La versión actual de IIS es la 6.0 para Windows Server 2003 e IIS 5.1 para Windows XP Professional. IIS 5.1 para Windows XP es una versión compacta del IIS que soporta sólo 10 conexiones simultáneas y sólo un sitio web. IIS 6.0 ha agregado soporte para IPv6.

Windows Vista viene con IIS 7.0 pre\_instalado. No limitará el número de conexiones permitidas pero limitará el flujo de tareas basándose en las solicitudes activas concurrentes, mejorando la usabilidad y el rendimiento en escenarios punto-a-punto (peer-to-peer).



## **2.2. Términos relacionados**

### **2.2.1. Servidor web**

Ordenador que hospeda información de distintos tipos: mensajes de correo, páginas WEB, archivos que pueden obtenerse mediante FTP, información de tipo GOPHER, NEWS, programas que permiten la charla (IRC e ICQ) etc. Así, se habla de servidores de correo cuando éstos almacenan los mensajes que llegan de otros destinatarios y se encarga de enviar los mensajes que elabore el remitente; servidores de Newsgroups cuando almacenan la información del tipo News que se envían; servidores FTP cuando ofrecen archivos que pueden obtenerse para uso personal; servidores WWW si en estos se encuentran publicados paginas WEB a las que accederemos mediante un navegador; servidores IRC e ICQ, que permiten la charla entre usuarios etc.

### **2.2.2. Portal WEB**

Un Portal Web es un conjunto de páginas que tienen como objetivo ofrecer al usuario, de forma fácil e integrada, el acceso a una serie de recursos y de servicios, entre los que suelen encontrarse buscadores, foros y otras tecnologías Web.

Las tecnologías Web implican el uso de servidores y clientes, así como del protocolo que permite la comunicación entre ambos, el denominado HTTP. Los

servidores son los encargados de brindar los servicios mientras que los clientes son los que consumen los mismos.

Para la construcción de un Portal Web es necesario que las páginas tengan la capacidad de interactuar con el usuario, de modo que muestre la información solicitada en tiempo real.

### **2.2.3. Página dinámica**

Las páginas que actualmente permiten realizar esta interacción con el usuario se denominan “páginas dinámicas”.

Una página es dinámica cuando se incluye funcionalidad como el acceso a una base de datos y usa lenguajes script como ASP y PHP, o lenguajes de programación tales como ASP .NET y JSP; aparte del simple HTML.

La tecnología JSP suministra una fácil manera de crear páginas Web dinámicas y simplifica la tarea de construir aplicaciones Web que trabajan con una gran gama de servidores Web, servidores de aplicaciones, navegadores y herramientas de desarrollo.

#### **2.2.4. Web services**

En nuestros días mucha información importante está almacenada en aplicaciones como sistemas de correo electrónico, y aún más recientemente en servicios de directorio. Microsoft habla sobre Universal Data Access (Acceso Universal a Datos) como una serie de manejadores e interfaces diseñadas para proveer una forma de conseguir acceder a este tipo de almacenamientos y más aún a datos como archivos de formato especiales, datos de posición geoespacial, datos científicos no estándar, etc.

Los servicios son puestos en la red y operan de manera cooperativa para dar soporte a uno o más procesos de negocios. En este modelo, una aplicación se convierte en un conjunto de servicios de usuario, negocios y datos que satisface las necesidades de los procesos de negocios o procesa su soporte.

Como los servicios están diseñados para el uso general y siguen lineamientos de interfaz publicados, pueden ser reutilizados y compartidos entre múltiples aplicaciones.

#### **2.2.5. Intranet**

Una Intranet es una red privada que la tecnología Internet usó como arquitectura elemental. Una red interna se construye usando los protocolos TCP/IP para comunicación de Internet, que pueden ejecutarse en muchas de las plataformas de hardware y en proyectos por cable. El hardware fundamental no

es lo que construye una Intranet, lo que importa son los protocolos del software. Las Intranets pueden coexistir con otra tecnología de red de área local.

Una intranet permitir a una compañía hacer negocios en línea, sin que estas transacciones **no necesiten nunca salir a Internet**. La información enviada a través de una Intranet alcanza su lugar exacto mediante los enrutadores, que examinan la dirección IP en cada paquete TCP/IP y determinan su destino.

Para **proteger la información corporativa**, y para asegurar que los piratas no perjudican a los sistemas informáticos y a los datos, las barreras de seguridad llamadas firewalls protegen a una Intranet de Internet. La tecnología firewall usa una combinación de enrutadores, servidores y otro hardware y software para permitir a los usuarios de una Intranet utilizar los recursos de Internet, pero evitar que los intrusos se introduzcan en ella.

### **2.3. Justificación de las herramientas utilizadas**

Entre las razones más importantes, está que la UEPMM, dispone de licencias Microsoft. Tomando en cuenta que Visual Studio 2008 es una herramienta muy eficiente para el desarrollo de aplicaciones Windows y Web que interactúan a una misma base de datos, y mejor aun si esta pertenece a la misma organización como es SQL Server 2008; la aplicación generada tendrá el soporte necesario tanto para el Frontend y el Backend utilizados al manejar políticas de seguridad y compatibilidad en su totalidad.

A continuación se detalla un cuadro comparativo de las herramientas utilizadas:

A nivel de Frontend

Cuadro 2.3: (Comparativo a nivel del Frontend)

<b>Visual Basic 2008 y C# 2008</b>	<b>Power Builder</b>
Conexión nativa con SQLServer.	XML Web DataWindow.
Creación de aplicaciones Windows y servicios Web mediante arrastrar-y-soltar, control estructurado de excepciones y sub procesamiento libre.	Componentes EAServer como Servicios Web.
Integración total con Microsoft .NET Framework y la Common Language Runtime (CLR).	Despliegue de JSPs hacia Enterprise Portal.
Nuevas herramientas de desarrollo para una conexión fácil con ASP.Net.	Cachés de conexión de EAServer para soporte a Unicode.

A nivel de Backend

Cuadro 2.4: (Comparativo a nivel del Backend)

<b>SQL Server 2008</b>	<b>ORACLE 10 G</b>
Optimiza automáticamente la base de datos.	Auto Afinación ( Grid Computing).
Herramienta sencilla de administración.	Entorno Web para la administración.
Cifrado de datos incorporado para lograr seguridad avanzada de datos.	Permite manejar gran variedad de carga de trabajo como:
Herramienta de generación de informes de usuarios finales.	<ul style="list-style-type: none"><li>▪ Servidores Sencillos</li><li>▪ Servidores con varios arreglos de disco</li><li>▪ Configuraciones Complejas</li></ul>

## CAPITULO III

### METODOLOGÍA

#### Metodología de desarrollo de sistemas

La metodología orientada a objetos será la utilizada para el desarrollo de la aplicación; para definir las diferentes clases con sus operaciones y atributos recopilados en el proceso que se sigue en el proceso de matriculas y pensiones; y en base a lo cual se podrá crear el modelo de datos que servirá de base para modelar el producto.

Un modelo O.O. es una abstracción cerrada semánticamente de un sistema; y este a su vez puede ser escrito por uno o más modelos, posiblemente desde distinto puntos de vista.

El modelado no es más que la construcción de un modelo a partir de una especificación, es una abstracción de algo, que se elabora para comprender ese algo antes de construirlo.

Para el desarrollo del proyecto se utilizará la metodología **MSF**, puesto que está es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.

MSF tiene las siguientes características:

- **Adaptable:** es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.
- **Escalable:** puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas a más.
- **Flexible:** es utilizada en el ambiente de desarrollo de cualquier cliente.
- **Tecnología Agnóstica:** porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación.

- *Modelo de Arquitectura del Proyecto:* Diseñado para acortar la planificación del ciclo de vida. Este modelo define las pautas para construir proyectos empresariales a través del lanzamiento de versiones.
- *Modelo de Equipo:* Este modelo ha sido diseñado para mejorar el rendimiento del equipo de desarrollo. Proporciona una estructura flexible para organizar los equipos de un proyecto. Puede ser escalado dependiendo del tamaño del proyecto y del equipo de personas disponibles.
- *Modelo de Proceso:* Diseñado para mejorar el control del proyecto, minimizando el riesgo, y aumentar la calidad acortando el tiempo de



entrega. Proporciona una estructura de pautas a seguir en el ciclo de vida del proyecto, describiendo las fases, las actividades, la liberación de versiones y explicando su relación con el Modelo de equipo.

- *Modelo de Gestión del Riesgo:* Diseñado para ayudar al equipo a identificar las prioridades, tomar las decisiones estratégicas correctas y controlar las emergencias que puedan surgir. Este modelo proporciona un entorno estructurado para la toma de decisiones y acciones valorando los riesgos que puedan provocar.
- *Modelo de Diseño del Proceso:* Diseñado para distinguir entre los objetivos empresariales y las necesidades del usuario. Proporciona un modelo centrado en el usuario para obtener un diseño eficiente y flexible a través de un enfoque iterativo. Las fases de diseño conceptual, lógico y físico proveen tres perspectivas diferentes para los tres tipos de roles: los usuarios, el equipo y los desarrolladores.
- *Modelo de Aplicación:* Diseñado para mejorar el desarrollo, el mantenimiento y el soporte, proporciona un modelo de tres niveles para diseñar y desarrollar aplicaciones software. Los servicios utilizados en este modelo son escalables, y pueden ser usados en un solo ordenador o incluso en varios servidores.

En conclusión la Metodología MSF se adapta a proyectos de cualquier dimensión y de cualquier tecnología.

### **3.1. Metodología MSF (Microsoft Solution Framework)**

El modelo del equipo de MICROSOFT SOLUTION FRAMEWORK (MSF por sus siglas en inglés) describe el enfoque de Microsoft a fin de organizar tanto a las personas como a sus actividades a fin de permitir el éxito del proyecto. El modelo define los grupos de roles, áreas funcionales, responsabilidades e instrucciones para que los miembros del equipo los dirijan de tal manera que puedan alcanzar sus metas únicas durante el ciclo de vida del proyecto.

A fin de maximizar el éxito de los proyectos de Información Tecnológica (IT) y operaciones durante todo el ciclo de vida de IT, la MICROSOFT SOLUTION FRAMEWORK y la Estructura de Operaciones de Microsoft (MOF) proporcionan orientación y prácticas probadas para planificar, construir, poner en marcha y dirigir soluciones en forma efectiva. Esta información se deriva de la experiencia ganada durante el desarrollo del software a gran escala de Microsoft y los proyectos de operación de servicios, la experiencia de los consultores de Microsoft, y las prácticas más comunes de la industria IT a nivel mundial. Se entrega en forma de reportes, guías, herramientas, plantillas, estudios de casos prácticos y cursos educativos. La orientación y prácticas están organizadas en dos conjuntos de conocimientos complementarios y bien integrados.

## **Componentes del MSF**

### ***Principios***

1. Promover comunicaciones abiertas.
2. Trabajar para una visión compartida.
3. Fortalecer los miembros del equipo.
4. Establecer responsabilidades claras y compartidas.
5. Focalizarse en agregar valor al negocio.
6. Permanecer ágil, y esperar los cambios.
7. Invertir en calidad.
8. Aprender de todas las experiencias.

### **Disciplinas**

#### **Gestión de Proyectos**

Es una disciplina que describe el rol de la gestión del proyecto dentro del modelo de equipo de MSF, y como permite mayor escalabilidad, desde proyectos pequeños a proyectos largos y complejos.

Se basa en:

- Planificar sobre entregas cortas.
- Incorporar nuevas características sucesivamente.
- Identificar cambios ajustando el cronograma.

## **Control de riesgos**

Diseñada para ayudar al equipo a identificar las prioridades, tomar las decisiones estratégicas correctas y controlar las emergencias que puedan surgir. Este modelo proporciona un entorno estructurado para la toma de decisiones y acciones valorando los riesgos que puedan provocar.

## **Control de Cambios**

Diseñada para que el equipo sea proactivo en lugar de reactivo. Los cambios deben considerarse riesgos inherentes y además deben registrarse y hacerse evidentes.

## **Modelos**

### **Equipo de Trabajo**

El Modelo de Equipo de Trabajo alienta la agilidad para hacer frente a nuevos cambios involucrando a todo el equipo en las decisiones fundamentales, asegurándose así que se exploran y revisan los elementos de juicio desde todas las perspectivas críticas.

Este modelo no es rígido, puede ser escalado dependiendo del tamaño del proyecto y del equipo de personas disponibles.



**Figura 3.1:** (Modelo del Equipo de Trabajo MSF)

## Proceso

El Modelo de Proceso de MSF, a través de su estrategia iterativa en la construcción de productos del proyecto, suministra una imagen más clara del estado de los mismos en cada etapa sucesiva. El equipo puede identificar con mayor facilidad el impacto de cualquier cambio y administrarlo efectivamente, minimizando los efectos colaterales negativos mientras optimiza los beneficios.

Este modelo ha sido diseñado para mejorar el control del proyecto, minimizando el riesgo, y aumentar la calidad acortando el tiempo de entrega.



**Figura 3.2:** (Modelo de Proceso de MSF)

### 3.2. Ejecución de la Metodología

Antes de ejecutar los diferentes procesos a seguir se describen los requisitos del sistema a desarrollar, estos requisitos fueron obtenidos bajo la realización de entrevistas con la directora y su cuerpo administrativo.

#### 3.2.1. Requisitos del Sistema

##### Generales

- Tanto para la aplicación Windows y Web, los sistemas deben controlar y restringir el acceso de usuarios. Tomando en cuenta que cada usuario debe tener una cuenta con claves distintas para cada aplicación, además el administrador del sistema podrá definir perfiles y estado de usuario.

- Cuando se ejecuten cobros sea este de matrícula o pensión, se debe registrar en forma automática fecha, hora y el usuario que realizó dicho cobro.
- La clave de los usuarios Windows y Web no puede ser conocida por el administrador de base de datos puesto que esta debe estar encriptada.

### Aplicación Windows

- Registrar, buscar, modificar estudiantes tomando en consideración los siguientes requisitos.
  - Datos del estudiante: Cédula, nombres y apellidos completos, sexo, nacionalidad, lugar y fecha de nacimiento, tipo de sangre, tipo de pensión, persona con quien vive, representante, estado del alumno.
  - Datos del Representante: Cédula, parentesco, nombres y apellidos, profesión ó ocupación, celular, dirección del domicilio, teléfono del domicilio, dirección del lugar de trabajo, teléfono del lugar de trabajo.
  - Datos del Padre: Cédula, nombres y apellidos, profesión ó ocupación, celular, dirección del domicilio, teléfono del domicilio, dirección del lugar de trabajo, teléfono del lugar de trabajo.
  - Datos de la Madre: Cédula, nombres y apellidos, profesión ó ocupación, celular, dirección del domicilio, teléfono del domicilio, dirección del lugar de trabajo, teléfono del lugar de trabajo.
  - Datos Varios: En este campo se podrá escribir en forma textual información relevante del estudiante.

- Cobrar matrícula generando automáticamente fecha, hora, tipo de matrícula, año lectivo, número de matrícula y costo de matrícula. Para realizar la matrícula tomar en consideración los siguientes pasos:
  - Buscar el estudiante por cédula ó código asignado por el sistema.
  - Ingresar los siguientes parámetros: año básico, paralelo, cédula, nombres y apellidos del responsable del pago.
  - Imprimir un comprobante de matricula que me permita respaldar la información antes mencionada é incluya los datos del usuario que realizo el cobro.
- Cobrar y eliminar pensiones, tomando en consideración los siguientes pasos:
  - Buscar el estudiante por cédula ó código asignado por el sistema.
  - Seleccionar las pensiones a ser pagadas.
  - Registrar meses cobrados, nombres y cédula de quien realizó dicho pago.
  - Para permitir eliminar una pensión se debe controlar que el usuario sea administrador ó director/a de la institución.
- Generar é imprimir reportes de pensiones de acuerdo a los siguientes requerimientos:
  - Cobradas por día que muestre los siguientes campos: código del estudiante, nombres y apellidos del estudiante, año básico, paralelo, mes cobrado y valor cobrado.
  - Cobradas y NO cobradas que muestre los siguientes campos: código del estudiante, nombres y apellidos del estudiante, año básico, paralelo, mes cobrado y valor cobrado; de acuerdo a los



parámetros mes y año lectivo. Al imprimir permita seleccionar si se desea o no imprimir en el reporte el valor cobrado. El reporte generado debe mostrar el total de estudiantes que si pagaron las pensiones para los parámetros seleccionados.

- Total cobradas que muestre los siguientes campos: mes cobrado y no cobrado, total del valor cobrado por mes y suma total; de acuerdo al año lectivo seleccionado.
- Cobradas y NO cobradas que muestre los siguientes campos: código del estudiante, nombres y apellidos del estudiante, año básico, paralelo, mes cobrado y valor cobrado; de acuerdo a los parámetros mes, año básico y año lectivo. Al imprimir permita seleccionar si se desea o no imprimir en el reporte el valor cobrado.
- Cobradas y NO cobradas que muestre los siguientes campos: código del estudiante, nombres y apellidos del estudiante, año básico, paralelo, mes cobrado y valor cobrado; de acuerdo a los parámetros mes, año básico, paralelo y año lectivo. Al imprimir permita seleccionar si se desea o no imprimir en el reporte el valor cobrado.
- Todas las pensiones cobradas por mes que muestre los siguientes campos: código del estudiante, nombres y apellidos del estudiante, año básico, paralelo, mes cobrado y valor cobrado; de acuerdo a los parámetros mes y año lectivo.

- Generar é imprimir reportes de estudiantes matriculados de acuerdo a los siguientes requerimientos:
  - Estudiantes matriculados en orden alfabético que muestre los siguientes campos: código generado por el sistema, cédula, nombres y apellidos, año básico, paralelo, valor, estado; de acuerdo al año lectivo seleccionado.
  - Estudiantes matriculados en orden alfabético por año básico y paralelo que muestre los siguientes campos: código generado por el sistema, lista, nombres y apellidos, año básico, paralelo, valor; de acuerdo al año lectivo seleccionado.
  - Estudiantes matriculados que muestre los siguientes campos: código generado por el sistema, lista, nombres y apellidos, año básico, paralelo, valor; de acuerdo a los siguientes parámetros año básico, paralelo y año lectivo seleccionado.
- Generar é imprimir reportes de estudiantes actuales de acuerdo a los siguientes requerimientos:
  - Estudiantes actuales en orden alfabético que muestre los siguientes campos: código, cédula, nombres y apellidos, año básico y paralelo; de acuerdo al año lectivo seleccionado.
  - Estudiantes actuales en orden alfabético que muestre los siguientes campos: código, cédula, nombres y apellidos, año básico y paralelo; de acuerdo a los siguientes parámetros año básico, paralelo, año lectivo seleccionado.

## Aplicación Web

- Cobrar matrícula generando automáticamente fecha, hora, tipo de matrícula, año lectivo, número de matrícula y costo de matrícula. Para realizar la matrícula tomar en consideración los siguientes pasos:
  - Buscar el estudiante por cédula ó código asignado por el sistema.
  - Ingresar los siguientes parámetros: año básico, paralelo, cédula, nombres y apellidos del responsable del pago.
  - La impresión de este comprobante es general desde la Web.
- Generar é imprimir reportes de estudiantes de acuerdo a los siguientes requerimientos:
  - Matriculados para un año lectivo seleccionado, que muestre los siguientes campos: código del estudiante, cédula, nombres y apellidos del estudiante, año básico, paralelo, valor cobrado, estado.
  - Actuales para un año lectivo seleccionado, que muestre los siguientes campos: Código, cédula, nombres y apellidos del estudiante, año básico y paralelo.
  - Matriculados para un año básico, paralelo y año lectivo seleccionado, que muestre los siguientes campos: código, lista, nombres y apellidos, año básico, paralelo, valor de pensiones que estos pagan.

### **3.2.2. Visión (Visión y Alcance Aprobados)**

Objetivo:

Obtener una visión del proyecto compartida, comunicada, entendida y alineada con los objetivos del negocio. Además, Identificar los beneficios, requerimientos funcionales, sus alcances y restricciones; y los riesgos inherentes al proceso.

#### **3.2.2.1. Documento Visión**

- Antecedentes y diseño

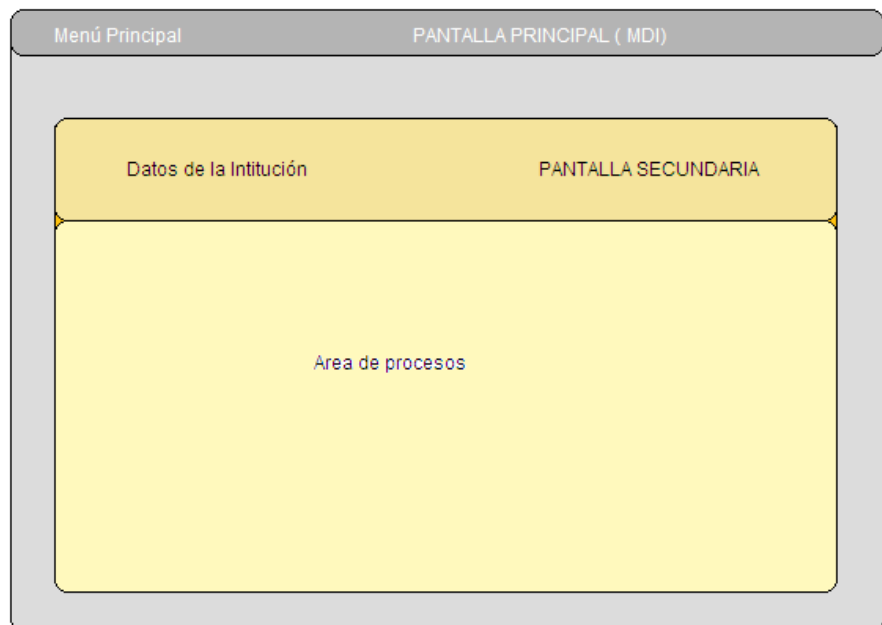
El sistema debe manejar dos ambientes Windows y Web, de acuerdo al alcance establecido.

- Criterios de diseño

Para el diseño Windows se tendrá en consideración los siguientes aspectos:

- El ingreso a realizar cualquier proceso, debe ser controlado y permitido de acuerdo al perfil del usuario
- Siempre que se ejecute cualquier proceso el sistema debe registrar en forma automática: usuario, fecha, hora; en que se ejecuto la transacción.
- Siempre que se desee buscar datos de un estudiante, sea este para modificar su información, cobrar una matricula o cobrar una pensión, se debe mostrar la fotografía del estudiante.
- Se tendrá una página principal MDI, donde se tiene un menú de acceso a las distintas pantallas de acuerdo al perfil que

tenga el usuario. Las pantallas secundarias que maneje el sistema siempre serán hijas de la pantalla MDI principal, estas deben tener el mismo diseño el cual indicara en la superior datos de la Institución completos, deben tener el mismo color, y manejar un estándar para la forma de botones.



**Figura 3.3:** (Diseño propuesto de interfaz Windows)

## Diseño conseguido:

Gerente de Sistemas (Wagner Eudimio Andrade Cedeño) ESCUELA PARTICULAR MIXTA MARIA MONTESSORI  
2008-07-13

Estudiantes --> DATOS DEL ESTUDIANTE

ESCUOLA PARTICULAR MIXTA  
MARIA MONTESSORI  
CHONE-MANABI  
Telf: 052-697-638

Datos Personales del Estudiante | Datos del Representante | Datos Personales del Padre | Datos Personales de la Madre | Datos Varios | Historia C

Código del Estudiante: 182

Cédula: [ ]

Primer Nombre: Jean

Segundo Nombre: Carlos

Primer Apellido: Andrade

Segundo Apellido: Solórzano

Sexo: Masculino

Nacionalidad: Ecuatoriana

Lugar y Fecha de Nacimiento: Shone, Día 27, Mes JULIO, Año 1999

Tipo de Sangre: [ ]

Tipo de Pensión: Seleccionar

Vive: Padres

Representante: Mamá

Estado del Alumno: Activo, Fecha: 2006-08-18 3:17:28

6  
Quinto Año Básico  
A

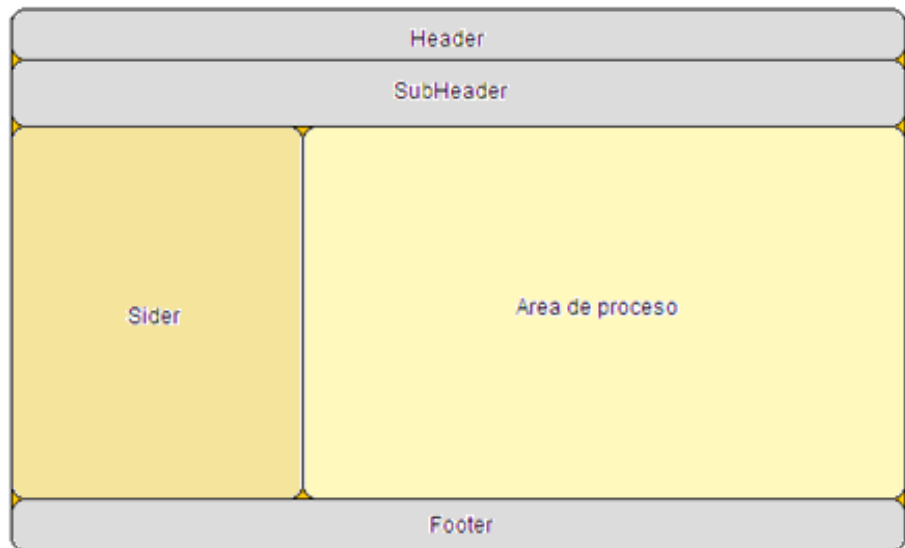
DATOS OBLIGATORIOS

GRABAR MODIFICAR NUEVO CANCELAR IMPRIMIR

**Figura 3.4:** (Diseño de interfaz Windows real)

Para el diseño Web se tendrá en consideración los siguientes aspectos:

- Se debe controlar la sesión de trabajo y su perfil
- Siempre que se ejecute cualquier proceso, se debe imprimir los datos del responsable, fecha y hora.
- La pantalla manejará cuatro espacios que estarán distribuidos de la siguiente forma:



**Figura 3.5:** (Diseño de interfaz Web)

**Header:** Se tendrán Link a páginas relacionadas con la institución, además que indicara el margen superior de la página web.

**SubHeader:** Se maneja como una segunda opción de publicidad y link de la institución. Será quien indique el margen para el área de trabajo de la página.

**Sider:** Tendremos el menú posible a navegar en la página dependiendo del perfil que se tenga.

**Área de Trabajo:** Maneja el área donde se pueden ejecutar los diferentes procesos. Ejm Seleccionar parámetros de un reporte.

**Footer:** Se tendrá Link a páginas relacionada con la institución, además que indicara el margen inferior de la página del área de trabajo y el margen inferior de la página Web.

### Diseño conseguido:



**Figura 3.6:** (Diseño de interfaz Web real)

#### 3.2.2.2. Documento detalle de Visión

- Beneficios, metas, objetivos y restricciones

Los beneficios del sistema son detallados en la justificación, al igual que la meta principal en el objetivo general y sus restricciones de principalmente de uso se las tiene en el siguiente tema perfiles de usuario.

- Perfiles de usuario



Estos perfiles de usuarios serán contemplados sea para la aplicación Windows ó Web.

**Director/a:** Tendrá autorización a realizar cualquier actividad académica, sea esta: Ingreso de estudiante, matricular, cobrar pensiones, eliminar pensiones, generar cualquier reporte.

**Administrador de Sistemas:** Bajo la autorización escrita del director/a, tendrá autorización a realizar cualquier actividad académica, sea esta: Ingreso de estudiante, matricular, cobrar pensiones, eliminar pensiones, generar cualquier reporte e incluso manipular la base de datos.

**Profesor/a:** Tendrá acceso únicamente a generar reportes, académicos sobre matriculas y pensiones sea cobradas o por cobrar.

**Secretaria/o:** Tendrá acceso a realizar matriculas, cobrar pensiones, ingresar estudiantes nuevos, modificar datos de estudiantes, y realizar cualquier reporte.

- Casos de uso

Los Casos de Uso no son parte del diseño, sino parte del análisis. De forma que al ser parte del análisis nos ayudan a describir qué es lo que es sistema debe hacer. Los Casos de Uso son los que identifican qué hace el sistema desde el punto de vista del usuario. Es decir, describen un uso del sistema y cómo este interactúa con el usuario.

Los casos de usos se pueden agrupar en diagramas, los diagramas no son lo importante.

Pero primero vayamos con lo realmente interesante.

Si lo primordial de los casos de uso (use case) no son los diagramas, entonces ¿que es lo importante? Lo realmente útil de los casos de uso es el documento que describe el caso de uso (use case), en este documento se explica la forma de interactuar entre el sistema y el usuario.

Pero lo más claro es que te presente uno. Este podría ser el caso de uso (use case) para realizar una matricula.

Cuadro 3.1: (Modelo de caso de uso real)

<b>Nombre:</b>	Realizar Cobro Matricula
<b>Autor:</b>	Wagner Andrade

<b>Fecha:</b>	13/07/2008
<b>Descripción:</b>	
Permite realizar una matricula en el sistema.	
<b>Actores:</b>	
Usuario Windows logeado.	
<b>Precondiciones:</b>	
El usuario debe haberse logeado en el sistema.	
<b>Flujo Normal:</b>	
<ol style="list-style-type: none"> <li>1. El actor pulsa sobre el botón para buscar datos del estudiante.</li> <li>2. El sistema muestra los datos del estudiante.</li> <li>3. El sistema permite que el actor seleccionar características de la matricula. Año básico, paralelo, y llenar datos del responsable del pago.</li> <li>4. El actor realiza la matrícula y la imprime si es necesario.</li> <li>5. El sistema comprueba la validez de los datos y los almacena.</li> </ol>	
<b>Flujo Alternativo:</b>	
5.. El sistema comprueba la validez de los datos, si los datos no son correctos, se avisa al actor de ello permitiéndole que los corrija	
<b>Pos</b>	<b>condiciones:</b>
La matricula ha sido almacenado en el sistema.	

**Nombre:** Nombre del caso de uso

**Autor:** Persona que creo el caso de uso

**Fecha:** Fecha que se creo el caso de uso

**Descripción:** Describe el proceso que se ejecuta en este caso de uso

**Actores:** Son aquellos que interactúan con el sistema.

**Precondiciones:** Son los hechos que se han de cumplir para que el flujo de evento se pueda llevar a cabo.

**Flujo:** Luego tenemos el flujo de eventos, que corresponde a la ejecución normal y exitosa del caso de uso (use case). Los flujos alternativos son los que nos permiten indicar qué es lo que hace el sistema en los casos menos frecuentes e inesperados.

**Pos condiciones:** Son los hechos que se ha de cumplir si el flujo de eventos normal se ha ejecutado correctamente.

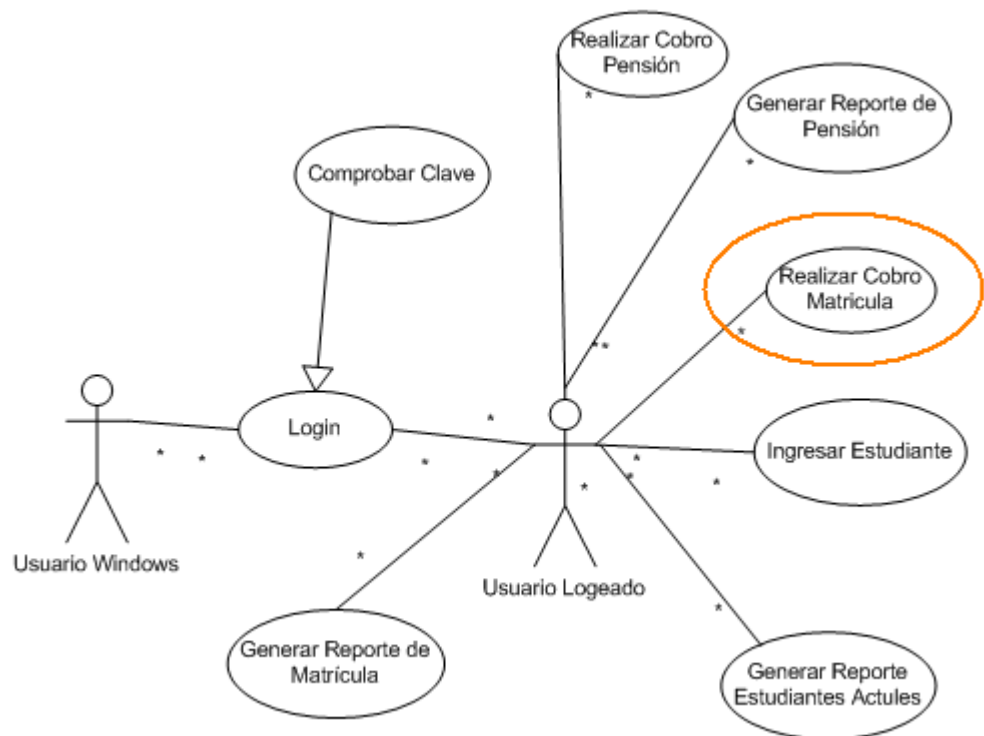
De forma que un caso de uso (use case) es un documento como el anteriormente presentado. Los casos de uso se pueden detallar más o menos dependiendo de la necesidad del problema.

Cuando empiezas a tener un número considerable de casos de uso como el anterior, no resulta nada fácil situarlos y relacionarlos. Entonces empiezas a tener la necesidad de una visión general del asunto, y ahora sí, es cuando los diagramas de casos de uso son de utilidad.

En los diagramas de casos de uso los muñecos son los actores y las pelotas son los documentos de casos de uso. Así que dibujas un muñeco por actor y una pelota por cada caso de uso (use case) y los enlazas con líneas cuando haya una relación entre ellos.

Con esto consigues una visión general de cómo los diferentes actores interactúan con los distintos casos de uso.

Para el caso de uso antes planteado se tendrá



**Figura 3.7:** (Modelo caso de uso)

- Requerimientos funcionales, no funcionales

**Los requerimientos funcionales:** Estos son los que describen lo que el sistema *debe* de hacer. Es importante que se describa el ¿Que? Y no el ¿Como? estos requerimientos al tiempo que avanza el proyecto de software se convierten en los algoritmos, la lógica y gran parte del código del sistema. Ej.:

- Panorama general: Este proyecto tiene por objeto crear un sistema Windows que permita realizar el cobro de matricula, pensiones, y reportes principales de estos cobros. Un sistema

Web que permita generar reportes para los cobros mencionados.

- Metas:

- ~ Realizar cobros de matrículas y pensiones de forma rápida.
- ~ Generar reportes rápidos y exactos de los cobros realizados.
- ~ Tener el control automático del cobro de pensiones.
- ~ Ingresar, modificar datos de estudiantes a la institución.

**Los requerimientos no funcionales:** Tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo

Atributos del Sistema:

- ~ Facilidad de uso
- ~ Tolerancia a fallas
- ~ Tiempo de respuesta, el rendimiento (en tiempo y espacio)
- ~ Metáfora de interfaz (interfaces de usuario)
- ~ Plataformas
- ~ Fiabilidad (robustez del sistema, disponibilidad de equipo)
- ~ Mantenimiento
- ~ Seguridad
- ~ Portabilidad
- ~ Estándares

- Requerimientos del sistema

## **Requerimientos mínimos**

- Microsoft Windows® Vista o anterior (Windows XP, 2008, 2003, 2000, NT4)
- Procesador Pentium® 133MHz (1GHz para Windows Vista o 300 MHz para Windows XP)
- 512 Mb de memoria RAM para Windows Vista, Windows Server 2008 o 128 Mb de memoria RAM para Windows XP, 2003, 2000 y NT4Tarjeta gráfica 16-bits 65536 colores - 640x480
- Ratón o dispositivo apuntador compatible
- Lector CD-ROM o DVD-ROM (distribución física)
- Internet Information Server, la versión que sea compatible de acuerdo al sistema operativo.
- SQL Server 2008
- Framework 3.0, de Visual Studio 2008
- Microsoft Visio 2007 (para poder leer los diseños)
- Microsoft Word 2007 (para poder leer los manuales)



- Plan de instalación

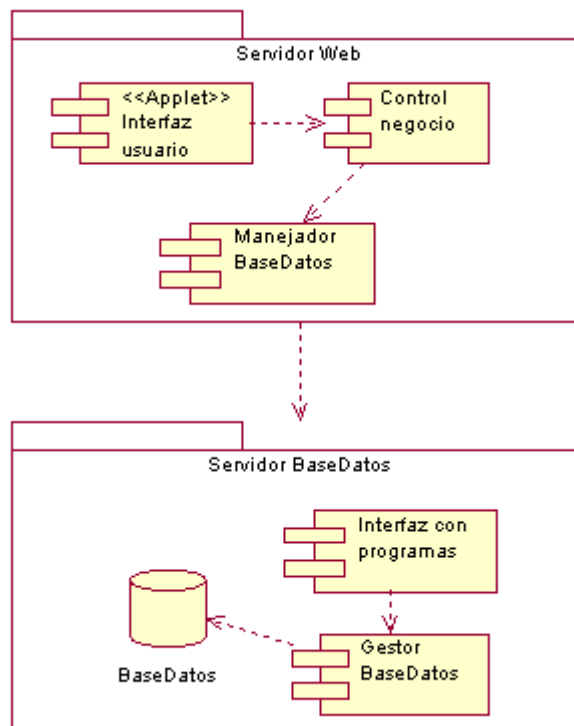
Cuadro 3.2: (Plan de instalación del sistema)

Equipo	Tareas de instalación y configuración
SS1 (Cliente Windows)	<ol style="list-style-type: none"> <li>1. Ejecute el instalador de Visual Studio 2008 en este equipo. Instale el Framework.</li> <li>2. Instale el sistema Windows UEPMM.</li> <li>3. Cree un ODBC que tenga el nombre ODBC_UEPMM y que se conecte a la base de datos UEPMM que estará en el servidor SS2.</li> <li>4. Copiar la carpeta Escuela Montessori en la unidad C. Este documento contiene las fotos de los estudiantes.</li> <li>5. Compruebe el funcionamiento del sistema.</li> </ol>

Equipo	Tareas de instalación y configuración
SS2 (Servidor de Base de Datos)	<ol style="list-style-type: none"> <li>1. Ejecute el instalador de SQL Server 2008 en este equipo. Instale el Servidor de Base de Datos.</li> <li>2. Ejecute el servidor y cargue la base de datos UEPMM y UEPMM_Datos</li> <li>3. Instale el Internet Information Server v.7.0</li> <li>4. Instale la aplicación Web que correrá sobre el servidor Web.</li> <li>5. Compruebe el funcionamiento del sistema Web.</li> </ol>
SS3 (Cliente Web)	<ol style="list-style-type: none"> <li>1. Solo debe tener instalado cualquier explorador Web que este soportado por Microsoft. Ej. Internet Explorer 7.0</li> </ol>

- Arquitectura lógica (Diagrama de componentes UML)

Muestra la dependencia entre los distintos componentes de software, incluyendo componentes de código fuente, binario y ejecutable. Un componente es un fragmento de código software (un fuente, binario o ejecutable) que se utiliza para mostrar dependencias en tiempo de compilación.

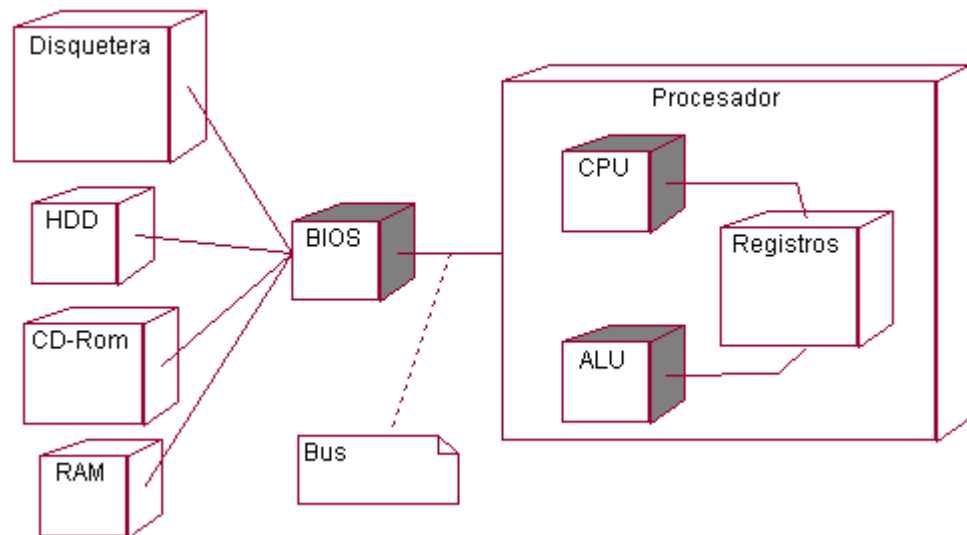


**Figura 3.8:** (Diagrama de Componentes)

- Arquitectura física (Diagrama de despliegue UML)

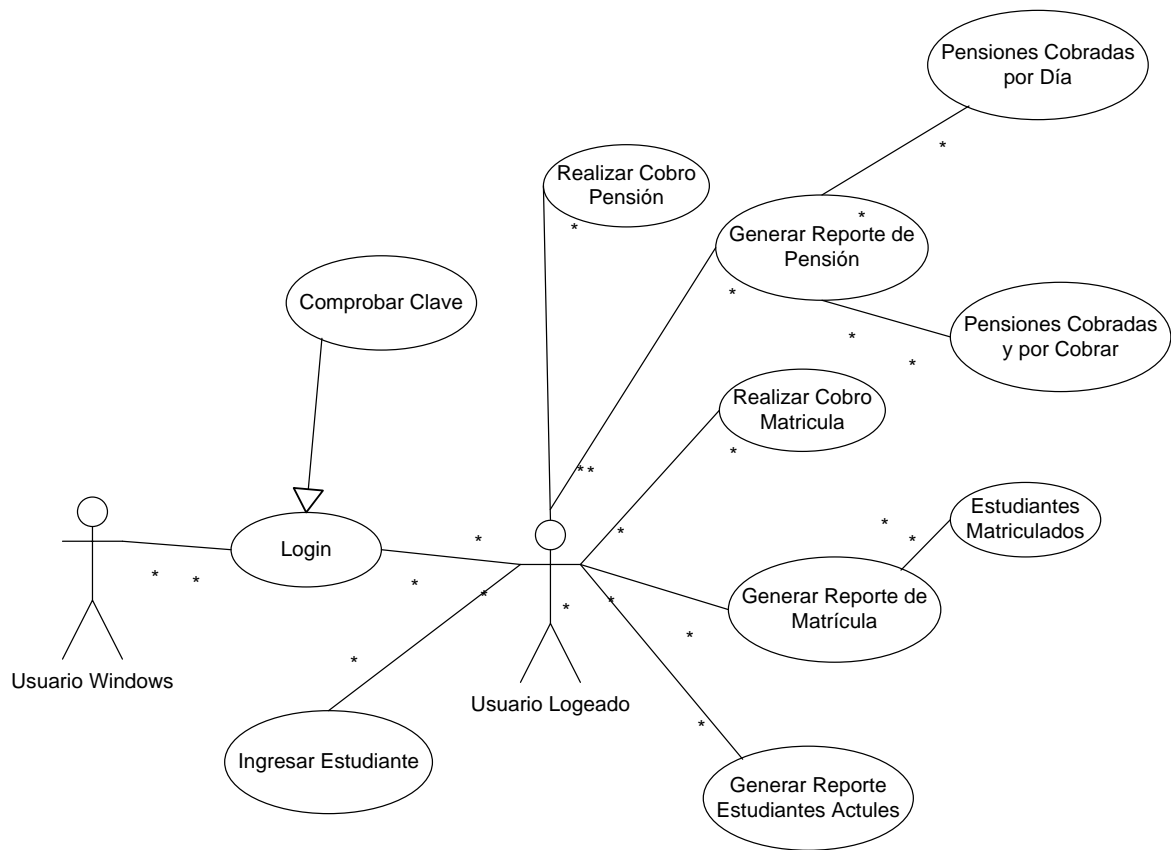
Muestra la configuración de los componentes hardware, los procesos, los elementos de procesamiento en tiempo de ejecución y los objetos que existen en tiempo de ejecución. En este tipo de diagramas intervienen nodos, asociaciones de comunicación, componentes dentro

de los nodos y objetos que se encuentran a su vez dentro de los componentes. Un nodo es un objeto físico en tiempo de ejecución, es decir una máquina que se compone habitualmente de, por lo menos, memoria y capacidad de procesamiento, a su vez puede estar formada por otros componentes.



**Figura 3.9:** (Diagrama de Plataforma o despliegue)

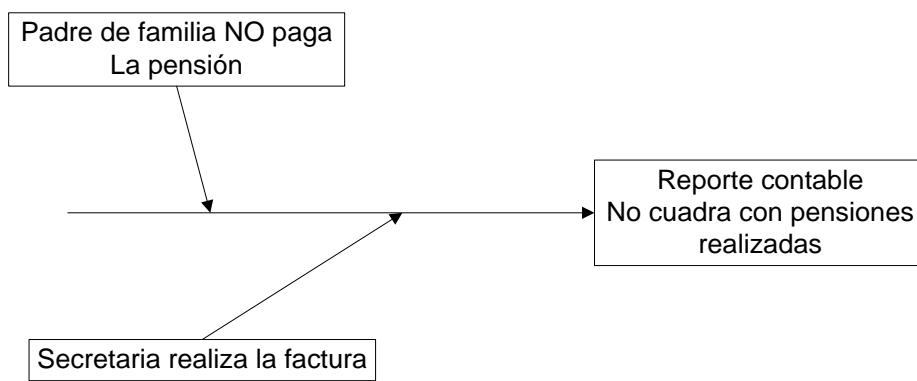
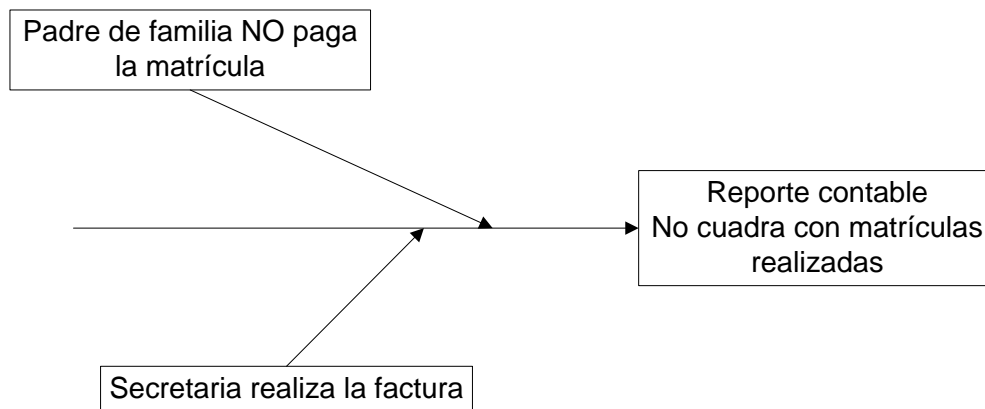
- Documento requerimientos funcionales (incluye script de pruebas)  
Descripción detallada de los requerimientos y características que componen cada caso de uso descrito en el documento Detalle de la Visión; indicando perfiles asociados, recursos del equipo de proyecto, riesgos, observaciones y script de pruebas.



**Figura 3.10:** (Caso de uso general)

### 3.2.2.3. Documento matriz de riesgos

- Identifica posibles riesgos acerca de los requerimientos y las acciones a tomar en cada escenario.



**Figura 3.11:** (Riesgos y acciones a tomar)

**3.2.2.4. Acta de aprobación de Visión**

### 3.2.3. Planeación (cronograma de proyecto aprobado)

Objetivo:

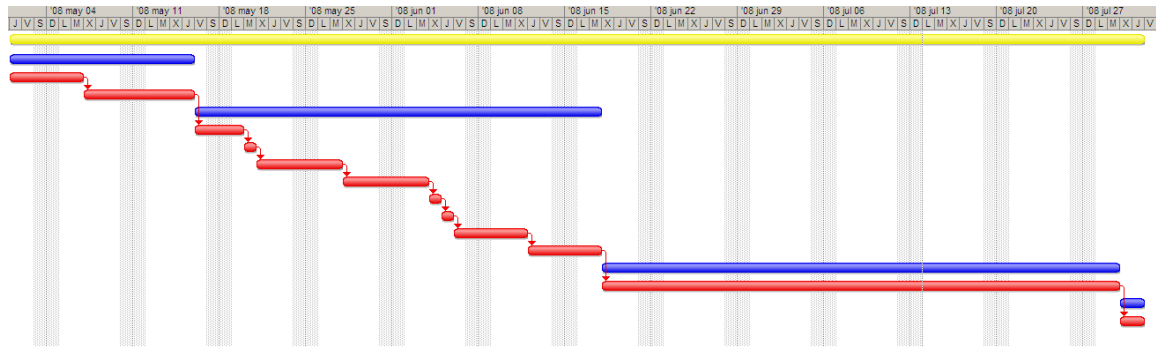
Obtener un cronograma de trabajo que cumpla con lo especificado en la fase de Visión dentro del presupuesto, tiempo y recursos acordados. Este cronograma debe identificar puntos de control específicos que permitan generar entregas funcionales y cortas en el tiempo.

Entregables

- Documento de Cronograma

Cuadro 3.3: ( Visión y Alcance Aprobados)

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
<b>Visión y Alcance Aprobados</b>	66 días	jue 08-05-01	jue 08-07-31	
<b><u>Documento Visión</u></b>	<b>11 días</b>	<b>jue 08-05-01</b>	<b>jue 08-05-15</b>	
Antedentes y diseño	4 días	jue 08-05-01	mar 08-05-06	
Criterios de diseño	7 días	mié 08-05-07	jue 08-05-15	3
<b><u>Documento detalle de Visión</u></b>	<b>23 días?</b>	<b>vie 08-05-16</b>	<b>mar 08-06-17</b>	
Beneficios, metas, objetivos y restricciones	2 días	vie 08-05-16	lun 08-05-19	4
Perfiles de usuario	1 día?	mar 08-05-20	mar 08-05-20	6
Casos de uso	5 días	mié 08-05-21	mar 08-05-27	7
Requerimientos funcionales, no funcionales	5 días	mié 08-05-28	mar 08-06-03	8
Requerimientos del sistema	1 día?	mié 08-06-04	mié 08-06-04	9
Plan de instalación	1 día?	jue 08-06-05	jue 08-06-05	10
Arquitectura lógica (Diagrama de componentes UML)	4 días	vie 08-06-06	mié 08-06-11	11
Arquitectura física (Diagrama de despliegue UML)	4 días	jue 08-06-12	mar 08-06-17	12
<b><u>Documento requerimientos funcionales</u></b>	<b>30 días</b>	<b>mié 08-06-18</b>	<b>mar 08-07-29</b>	
Documento	30 días	mié 08-06-18	mar 08-07-29	13
<b><u>Documento matriz de riesgos</u></b>	<b>2 días</b>	<b>mié 08-07-30</b>	<b>jue 08-07-31</b>	
Documento	2 días	mié 08-07-30	jue 08-07-31	15



**Figura 3.12:** (Diseño de tiempos de la visión y alcance aprobados)

- Acta de aprobación de Cronograma

### 3.2.4. Desarrollo(alcance completo)

Objetivo:

Obtener iterativamente de la mano de la fase de Planeación y de la de Estabilización versiones del producto entregables y medibles que permitan de cara al cliente probar características nuevas sucesivamente. Esto incluye los ajustes de cronograma necesarios.

Entregables:

- Fuentes y ejecutables (Según lo acordado)
- Documentos Manuales técnicos, de usuario y de instalación si es necesario.
- Acta de finalización de desarrollo



### **3.2.5. Estabilización( versión aprobada)**

Objetivo:

Obtener una versión final del producto probada, ajustada y aprobada en su totalidad

Entregables:

- Documento Registro de pruebas
- Acta de aprobación de Versión Aprobada

### **3.2.6. Instalación(Entregas)**

Objetivo:

Entregar (instalar) al cliente el producto finalizado en su totalidad. Como garantía se han superado con éxito las etapas anteriores.

Entregables:

- Conjunto de archivos (ejecutables, directorios, archivos varios, bases de datos, scripts, instaladores, manuales, licencias, entre otros) propios del producto que permitan su instalación y correcto funcionamiento.
- Acta de Entrega y Finalización de Proyecto.

### **3.2.7. Soporte(entrega ajustada)**

Objetivo:

Brindar soporte y garantía al producto durante el tiempo estipulado en el contrato; registrando los reportes de soporte y mantenimiento recibidos, así como los ajustes y versiones ajustadas obtenidas. Esto sólo será válido para ajustes que estén dentro de lo descrito en los documentos de la fase de Visión.

Entregables:

Documento de registro de reportes de soporte y mantenimiento y ajustes hechos

## CAPITULO IV

### CONCLUSIONES Y RECOMENDACIONES

#### 4.1. Conclusiones

- El uso de la metodología MSF optimiza el desarrollo de proyectos orientados al desarrollo Windows y páginas Web dinámicas ya que para realizarla es necesario primero realizar el análisis que sirva como base para el desarrollo de sistemas de pequeña, mediana ó gran escala.
- La creación de dos ambientes Windows y Web, brinda una gran cantidad de beneficios institucionales, pues al tener centralizada la información y poder compartir recursos de esta, nos permite ejecutar de forma eficiente los procesos de la institución y poder realizar sus diferentes reportes desde cualquier PC, en nuestra Intranet.
- Utilizar el servidor IIS y su plataforma Microsoft, resulta fácil de usar y con un amplio soporte para distintos exploradores de internet como: Mozilla Firefox (es 100 % gratuito, es el que yo utilizo y vale la pena descargarlo), Opera, Internet Explorer, Netscape, FineBrowser, Safari. Lo que permite mejorar la calidad y ampliar el uso de nuestro sistema.

- Crear aplicaciones Web dinámicas permite tener software de calidad, seguro, rápido, y posible de ser utilizado desde cualquier PC, conectado a la red lo que nos permite brindar un mejor servicio al cliente.
  
- La tecnología WEB ha tenido un gran impacto para solucionar problemas de centralización de la información y permite ser usada por los distintos lenguajes de programación.
  
- Los sistemas informáticos son una solución ágil para una empresa en constante crecimiento, y dispuestas abrirse a la tecnología.

## 4.2. Recomendaciones

- Utilizar siempre tecnología de punta principalmente en el desarrollo de Software, aunque no lo parezca, resulta muy fácil cambiar de tecnología de una versión a otra pues para nuestro caso el Visual Studio 2008 permite migrar casi de forma automática de una versión a otra aun si es que el proyecto es muy antiguo, como es el caso de aplicaciones creadas en Visual Studio 6.0.
  
- Desarrollar las aplicaciones con una activa participación de los usuarios, de manera que estos se sientan parte del éxito del proyecto.
  
- Desarrollar sistemas personalizados que se ajusten a las necesidades y requerimientos reales de la empresa.
  
- Usar la tecnología Web en los distintos lenguajes de programación para que de esta manera permita una eficaz interacción e integración del conjunto de datos de manera que sea un sistema de información parametrizado y abierto a los cambios e innovaciones de la tecnología y del mercado.

- Desarrollar otros módulos como el control de notas desde la Web, que permita el ingreso y reporte de notas tanto para personal administrativo, padres de familia, y estudiantes.

## Glosario

- **DCOM.-** Distributed Component Object Model, Modelo de objetos de componentes distribuido.
- **UEPMM.-** Unidad Educativa Particular Maria Montessori.
- **MSF.-** Microsoft Solution Framework.
- **SCM.-** Services Control Manager, Administrador de control de servicios
- **API.-** Application Programming Interface, Interfaz de Programación de Aplicaciones.
- **TCP.-** Transmisión Control Protocol, Protocolo de Control de Transmisión.
- **IPX.-** Internetwork Packet Exchange, Intercambio de paquetes entre redes.
- **HTTP.-** HyperText Transfer Protocol, Protocolo de Transferencia de hipertexto.
- **ASP.-** Application Server Providers, Proveedores de Servidor de Aplicaciones.
- **JSP.-** Java Server Page, Páginas de Servidor Java
- **PHP.-** Personal Home Page
- **CLR.-** Common Language Runtime, Entorno de ejecución de la plataforma.
- **XSD.-** Xml Schema Definition, Definición de Esquemas Xml.
- **CORBA.-** Common Object Request Broker Architecture, Arquitectura de intercambio de solicitudes de objetos comunes.
- **JIT.-** Just-in-Time, Justo a Tiempo.

## Bibliografía

- GEOCITIES, *Diseño de Aplicaciones Tres Capas* [en línea]. [marzo 2004]  
Disponible en Web: < <http://www.geocities.com/trescapas/TresCapas.htm>>.  
Ultimo acceso: 20 de junio 2008.
  
- TELYSE, Área Técnica. *Manual JAVASCRIPT*. [marzo 2004].  
Disponible en Web:  
< <http://www.telyse.net/telyse/areatecnica/manuales/js/capitulo2.html>>.  
Ultimo acceso: 20 de junio 2008.
  
- ABC DATOS Programas Y Tutoriales En Castellano, *Programación : ASP .NET*. Disponible en Web:  
< <http://www.abcdatos.com/tutoriales/programacion/asp.net.html> >  
Ultimo acceso: 18 de julio 2008.
  
- EMAGISTER La Guía Inteligente de Formación, *Cursos Visual C# .NET*[Agosto 2006].  
Disponible en Web:  
<[http://www.emagister.com/index\\_buscadore.cfm?action=search&frmStrBusqueda=Visual+C%23+2008](http://www.emagister.com/index_buscadore.cfm?action=search&frmStrBusqueda=Visual+C%23+2008) >.  
Ultimo acceso: 20 de enero 2008.



- WEBTALLER, *Las Tecnologías de las Páginas Web* [en línea]. [febrero 2007].  
Disponible en Web:  
<[http://www.webtaller.com/maletin/articulos/las\\_tecnologias\\_de\\_las\\_paginas\\_web.php](http://www.webtaller.com/maletin/articulos/las_tecnologias_de_las_paginas_web.php)>.  
Ultimo acceso: 18 de marzo 2008.
  
- WEB URBANO. *WEB SERVICES*. [Marzo 2006]  
Disponible en Web: < <http://web.urbano.com.mx/soporte/?secc=1>>  
Ultimo acceso: 18 de marzo 2008.
  
- MONOGRAFÍAS. *Visión Global INTRANET*. [Abril 2005]  
Disponible en Web:  
<<http://www.monografias.com/trabajos12/intrants/intrants.shtml>>  
Ultimo acceso: 15 de abril 2007.
  
- WNTMAG. *INTRANET Resources*. [Enero 2002]  
Disponible en Web:  
<[http://www.wntmag.com/atrasados/1996/02\\_oct96/intranet.html](http://www.wntmag.com/atrasados/1996/02_oct96/intranet.html)>  
Ultimo acceso: 15 de mayo 2008.
  
- Booch, Jacobson, Rumbaugh, El Proceso Unificado de Desarrollo de Software. (1999). Editorial Addison Wesley.

- MONOGRAFÍAS. *El Desarrollo De Sistemas De Información Empleando El Lenguaje De Modelado Unificado UML.*

Disponible en Web:

<<http://www.monografias.com/trabajos16/lenguaje-modelado-unificado/lenguaje-modelado-unificado.shtml#PROCESO>>

Ultimo acceso: 15 de enero 2008.

- MSDN, Canchala Armando. *Un Ejemplo Sencillo Sobre Modelado de un Proyecto* [Abril 2006]

Disponible en Web:

<[http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ\\_2295/default.aspx](http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_2295/default.aspx)>

Ultimo acceso: 15 de junio 2008.

- MSDN, Visual Studio 2008.

#### Otras direcciones web visitadas

- **Web Services Developer Center.** <http://msdn.microsoft.com/webservices/>.
- **Web Services Activity.** <http://www.w3.org/2002/ws/>.
- **Web Services Architect.** [www.webservicesarchitect.com/](http://www.webservicesarchitect.com/).
- **Web Services Journal .** [www.sys-con.com/webservices/](http://www.sys-con.com/webservices/).
- **Microsoft MSDN.** <http://www.microsoft.com/msdn>
- **Microsoft MSDN Español**  
<http://www.microsoft.com/spanish/msdn/ecuador/default.asp>
- **Miscrosoft MSDN WebServices Site** <http://msdn.microsoft.com/webservices/>
- **Desarrollador Cinco Estrellas** <http://www.microsoft.com/spanish/ms>
- **.NET Achitecture Center**  
<http://msdn.microsoft.com/architecture/soa/default.aspx>  
<http://www.microsoft.com/spanish/msdn/arquitectura/default.asp>
- **Tutorial de procedimientos de Microsoft**  
<http://www-106.ibm.com/developerworks/webservices/newto/>
- **Microsoft Patterns & Practices**  
<http://msdn.microsoft.com/practices/>
- **Microsoft Application Architecture for .net: Designing Applications and Services.** <http://msdn.microsoft.com/practices/>

**HOJA DE LEGALIZACIÓN DE FIRMAS**

**ELABORADO POR:**

---

Wagner Eudimio Andrade Cedeño

**DECANO DE LA FACULTAD DE LA INGENIERÍA EN SISTEMAS E  
INFORMÁTICA**

---

Ing. Ramiro Delgado

Sangolquí, 28 de julio del 2008