PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR

SEDE AMBATO

ESCUELA DE INGENIERÍA EN SISTEMAS

DISERTACIÓN DE GRADO PREVIA LA OBTENCIÓN DEL TITULO DE INGENIERA DE SISTEMAS

"DESARROLLO DE UNA APLICACIÓN WEB PARA EL REGISTRO Y CONSULTAS DE NOTAS EN LA ESCUELA DE INGENIERÍA DE SISTEMAS DE LA PUCESA UTILIZANDO LA HERRAMIENTA VISUAL STUDIO.NET"

CATALINA ELIZABETH MORALES PERRAZO

DIRECTOR DE LA DISERTACION: ING. VICTOR CHUNCHA

AMBATO 2005 - 2006

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR
PROGRAMAS ACADEMICOS AMBATO
BIBLIOTECA

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR
SEDE AMBATO
SECRETARIA
ESCUELA DE INGENIERIA
DE SISTEMAS
17-I-2006

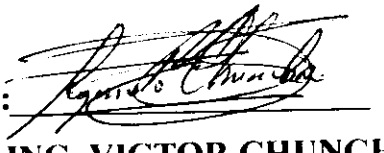
PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR

SEDE AMBATO

ESCUELA DE INGENIERÍA DE SISTEMAS

DISERTACIÓN DE GRADO PREVIA LA OBTENCIÓN DEL TITULO DE INGENIERA DE SISTEMAS

"DESARROLLO DE UNA APLICACIÓN WEB PARA EL REGISTRO Y CONSULTAS DE NOTAS EN LA ESCUELA DE INGENIERÍA DE SISTEMAS DE LA PUCESA UTILIZANDO LA HERRAMIENTA VISUAL STUDIO.NET"

DIRECTOR : 

ING. VICTOR CHUNCHA

CATALINA ELIZABETH MORALES PERRAZO

AUTORIA DE TESIS

Yo, Catalina Elizabeth Morales Perrazo declaro que la presente investigación enmarcada en el diseño del presente Plan y que presente como Plan de Disertación es absolutamente original, autentico y personal. En virtud declaro que el contenido de la presente es de exclusiva responsabilidad legal y académica mía.

Catalina Elizabeth

ÍNDICE

CAPITULO I

1. PROYECTO DE INVESTIGACIÓN 1

1.1. Planteamiento del problema 2

1.2. Objetivos 3

1.3. Hipótesis 4

1.4. Importancia y Justificación 4

1.5. Plan Operativo 6

CAPITULO II

2. MARCO TEÓRICO

2.1. Crear y mantener una Base de Datos 7

2.1.1. Bases de Datos 7

2.1.1.1. Crear una Base de Datos 9

2.1.1.2. Modificar y Eliminar una Base de Datos 10

2.1.2. Tablas 11

2.1.2.1. Diseñar Tablas 11

2.1.2.2. Crear y modificar una tabla 12

2.1.2.2.1. Crear una tabla 12

2.1.2.2.2. Modificar una Tabla 14

2.1.2.2.3. Eliminar una tabla 14

2.1.3. Normalización 14

2.1.4. Triggers 17

2.1.5. Definición de Usuarios 18

2.1.6. Roles y privilegios 19

2.1.6.1. ROLES 21

2.1.6.2.PRIVILEGIO 22

2.2. Tener Acceso y Cambiar Datos 23

2.2.1. Agregar Datos 23

2.2.2. Cambiar Datos 24

2.2.3. Eliminar Datos 24

2.3. Tecnologia .Net 24

2.3.1. Microsoft .Net 26

2.3.1.1. Common Language runtime(CLR) 31

2.3.1.2. Microsoft Intermediate Languaje (msil) 32

2.3.1.3. Metadatos 32

2.3.1.4. Ensamblados 34

2.3.2. Visual C# 35

2.3.3. Visual J # 40

2.3.4. Visual C++ 46

2.3.5. Visual Basic .NET 48

2.3.5.1.Proyectos Visual Basic 50

2.3.5.2.Programación Web 51

2.3.5.3. El acceso a datos en la Web 52

2.3.5.4.Servicios web XML 54

CAPITULO III

3. DESARROLLO DE LA APLICACIÓN 59

3.1.1. Análisis del Diseño 59

3.1.2. Análisis de la Base de Datos 60

3.1.3. Implementación de la Base de Datos 60

3.1.4. Diseño de la Interfaz 62

3.1.5. Desarrollo de la Aplicación Web en Visual Studio. Net 66

3.1.6. Implementación de la Aplicación 67

CAPITULO IV

4. VALIDACIÓN DEL PROYECTO 78

4.1. COMPROBACIÓN DE LA HIPÓTESIS 79

CAPITULO V

5.1. CONCLUSIONES Y RECOMENDACIONES 80

5.1.1.CONCLUSIONES 80

5.1.2. RECOMENDACIONES 81

GLOSARIO DE TERMINOS 82

MANUAL DE USUARIO 84

MANUAL DE INSTALACION 87

DIAGRAMAS

ITERACIÓN 63

SECUENCIA 63

PROFESOR 64

ESTUDIANTE 64

ADMINISTRADOR 65

MODELO ENTIDAD RELACIÓN 65

DICCIONARIO DE DATOS 69

TABLAS

Tabla # 1. Campos De las tablas 12

Tabla # 2. Base de Datos 14

GRÁFICOS

Grafico # 1 Componentes de Visual Stdio.Net 25

Grafico # 2 Tecnología. NET 27

Grafico # 3 Plataforma. NET 28

Grafico # 4 Arquitectura del . NET Framework 30

Grafico # 5 Arquitectura de J# 44

Grafico # 6 Proyecto en Visual Basic 51

Grafico # 7 Arquitectura Básica de ADO.NET 54

Grafico # 8 Esquema Gráfico de un WEB service 56

Grafico # 9 Modelos de Servicios Web 58

DEDICATORIA

Este proyecto se la dedico primeramente a Dios, ya que sin Él nada podemos hacer. Él es quien nos concede el privilegio de la vida y nos ofrece lo necesario para lograr nuestras metas. Gracias de todo corazón por permitirme estar aquí, por las pruebas que me hacen crecer como persona y ser humano y me permiten dar lo mejor de mí.

También le dedico este proyecto a mis padres Carlos, Bertha y a mis hermanos Nelson y Fernanda, porque ellos siempre estuvieran junto a mi en las buenas y en las malas; me educan, me aconsejan, me imparten valores para conducirme correctamente y me ofrecen el sabio consejo en el momento oportuno.

Este trabajo está hecho con todo mi amor y toda mi dedicación, lo cual produce una gran satisfacción en poder servir a quien así lo requiera.

Catalina Elizabeth

AGRADECIMIENTO

Existen unas personas bien cerca de nosotros que en la mayoría de las ocasiones nos brindan su amistad y su cooperación justo en el momento cuando lo necesitamos. Y en esta ocasión deseo expresar mi gratitud a todas estas personas: GRACIAS: a dios , a mis padres, hermanos, a todos los que de alguna forma estuvieron a mi lado con su apoyo compresión y amor.

Aquellos que influyeron en mi compartamiento, sus vidas conmigo forjando mi educación con los valores y conocimientos. Nunca podré pagar tanta riqueza que he recibido a lo largo de mi vida y siempre estaré profundamente agradecida.

También agradezco al Tutor de mi disertación ya que sin la guía de él no hubiera terminado este proyecto.

Una vez mil gracias a todos.

Catalina Elizabeth

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
PROGRAMAS AMBATO
BIBLIOTECA

RESUMEN

Con la aparición del plataforma .NET se ha iniciado una nueva era en el campo de la programación de aplicaciones que conducirá al Internet de nueva generación ASP.NET, una parte de la plataforma .NET de Microsoft, es una estructura de programación revolucionaria que permite el desarrollo de aplicaciones Web.

La Disertación esta proyectada de manera que pueda usarse como complemento de cualquier texto; servirá un libro de consulta y material de apoyo para los estudiantes.

Cada capítulo comienza con una clara exposición de definiciones acompañados de un material ilustrativo de cada una de las herramientas de Visual Studio . Net .

El primer capítulo corresponde a la descripción del problema, la hipótesis, importancia y objetivos que se consideraron para poder desarrollar.

En el segundo capítulo se detalla la investigación: En la base de datos se puede agregar, borrar, eliminar, modificar los datos además también se detalla, la nueva herramienta Visual Studio .Net en la cual se da a conocer, conceptos estructuras, de cada uno de los componentes como de: Visual Basic.NET, C#.NET, J#.NET, C++.NET

En el tercer capítulo se desarrolla la aplicación en Visual Basic .NET con Sql Anywhere 5.0 de una pagina web en la cual tanto los estudiantes, profesores podrán acceder a los datos, al igual que el administrador podrá actualizar la base datos.

En el cuarto capítulo se encuentra las validación del proyecto y la comprobación de la Hipótesis.

En el Quito capitulo se detalla las conclusiones, recomendaciones que se logran en el proceso de desarrollo.

Por ultimo tenemos el manual del usuario en el cual se detalla paso a paso como se debe ingresar al sistema.

También se encuentra el glosario de Términos y el manual de Instalación del Sistema.

CAPITULO I

1. PROYECTO DE INVESTIGACIÓN

Hoy en día todo gira acorde con la tecnología actual es por ello que es de vital importancia para las Instituciones tanto de índole pública como privada ir actualizándose en todas las actividades que estas realizan a diario, y que mejor por medio de aplicaciones actualizadas como son las herramientas de desarrollo de la Empresa Microsoft conocidas como .NET.(Visual Basic, Visual C#, Visual J#, Visual C++)

NET es el nombre dado a una estrategia nueva: Un proyecto original para la creación de aplicación de la nueva década, es toda una nueva arquitectura tecnológica, desarrollada por Microsoft para la creación y distribución del software como un servicio. Esto quiere decir que mediante las herramientas de desarrollo proporcionadas por esta nueva tecnología, los programadores podrán crear aplicaciones basadas en servicios para la Web. Por medio de la creación de una aplicación realizada en Visual Basic .NET, se proveerá en la Escuela de Ingeniería en Sistemas de la PUCESA de una herramienta de apoyo que contribuirá al mejor desempeño de las actividades de los docentes en la Institución y al manejo rápido, eficaz, eficiente de información que requieren estudiantes, profesores y administrativos de la Universidad.

Además se proveerá al los docentes y alumnos en la Escuela de Ingeniería en Sistemas de la PUCESA de la documentación necesaria para realizar aplicaciones en la plataforma .Net basándose en un amplio estudio de la plataforma .Net y sus herramientas de desarrollo. Actualmente es necesario que el programador sepa aprovechar las bondades de la informática es por eso que debemos enfocamos a las herramientas de software actualizadas por las ventajas que estas prestan, una de esas ventajas es la utilización de estas aplicaciones mediante la red más grande del mundo que es Internet, utilizadas hoy en día por la mayoría de personas.

1.1. PLANTEAMIENTO DEL PROBLEMA

¿La Escuela de Ingeniería en Sistema de la PUCESA no cuenta con un software adecuado para el ingreso de notas utilizando herramientas.net mediante el Internet en el periodo 2003 – 2004?.

PROBLEMATIZACIÓN

- Carencia de estudios de la herramientas.net
- Los profesores no utilizan herramientas.net dentro de las cátedras.
- El no conocimiento de los profesores y estudiantes de las ventajas y servicios la herramientas de desarrollo.net.
- El sistema de notas en secretaria se lleva en forma tradicional utilizando el computador como un medio de respaldo de la información.

DELIMITACIÓN

En el proyecto se desarrollará un Sistema de ingreso de notas mediante una aplicación de las herramientas.net para la Escuela de Ingeniería de Sistemas de la PUCESA en el periodo marzo 2004 – septiembre 2004 constará y tratara sobre la siguiente temática:

- Al concluir la disertación de grado, en la Escuela de Ingeniería en Sistemas de la PUCESA contará con un soporte académico documental y digital para la utilización de las herramientas de Visual Estudio.Net y a su vez con una aplicación que estará realizada con un motor de base de datos SQL Anywhere 5.0 y Visual Basic.Net la misma que estará enfocada al control de notas de los estudiantes de la Escuela de Ingeniería en Sistemas de la PUCESA, ayudando notablemente a la actualización de los registros académicos de profesores y alumnos en la Escuela de Ingeniería en Sistemas de la PUCESA

- Tomando en consideración que una de las formas más fáciles para el desarrollo intelectual de los integrantes (Profesores, alumnos, administrativos) de la PUCESA es mediante el acceso a cursos y que mejor que los mismos sean

dictados por futuros profesionales que se han formado en la PUCESA, luego de la finalización de la disertación se dictará un curso relacionado con la misma. Este curso tratara a cerca del funcionamiento de las Herramientas de Visual Studio. Net así como también se hará una demostración practica mediante la utilización de Visual Studio.Net.

1.2. OBJETIVOS

OBJETIVO GENERAL

Construir un software para el ingreso de notas y consultas de los alumnos de la Escuela de Ingeniería en Sistemas de la PUCESA utilizando las herramientas.net en el periodo 2003 - 2004

OBJETIVOS ESPECIFICOS

- Darles a conocer sobre la funcionalidad y servicios que presta la herramientas.net
- Ampliar la visión de los docentes sobre nuevas herramientas.Net que se encuentran en el mercado.
- Realizar cursos de capacitación para conocer las ventajas y desventajas que ofrece la herramienta.net
- Dotar de un sistema de ingreso de calificaciones de la Escuela de Ingeniería en Sistemas de la PUCESA mediante la tecnología actual utilizando las herramientas.net.

1.3. HIPÓTESIS

Con la implementación de una aplicación Web para el ingreso de notas de la Escuela de Ingeniería en Sistemas de la PUCESA permitirá mayor facilidad al acceso de calificaciones por parte de los docentes.

1.4. IMPORTANCIA Y JUSTIFICACIÓN

A nivel nacional el estudio será la base para el desarrollo de futuros sistemas en herramientas.net tanto en las empresas públicas y privadas.

A nivel local al realizar el estudio de las herramientas.net, se mejorarán los sistemas tradicionales, reduciendo distancia y ganando tiempo con las transacciones.

La creación de este sistema nos permitirá modernizar a la Institución y resolverá el problema actual que es el ingreso de notas mediante el Internet, con esta nueva alternativa se persigue la eficiencia como parte vital para el progreso de la Institución la Escuela de Ingeniería de Sistemas de la PUCESA será la pionera en dar un ejemplo a nivel de universidades y a nivel de escuelas ya que presentará un sistema de alta calidad como es el ingreso de notas mediante el Internet utilizando las herramientas.net, y de esta forma agilitara el ingreso de notas el mismo que ayudara a los docentes, secretarias y estudiantes a disminuir la latencia de estos sistemas.

En al ámbito económico justifica este estudio ya que el factor económico no es de vital importancia por que la herramienta.net si existe en la universidad.

En el ámbito Educativo justifica este estudio por que en la Universidad existe docentes que conocen herramientas nuevas que están en el mercado y mediante esto podemos llegar a ser unos buenos profesionales para el futuro.

METODOLOGÍAS DE INVESTIGACIÓN

Las metodologías a utilizarse nos permitirá ordenar las actividades del proceso investigativo:

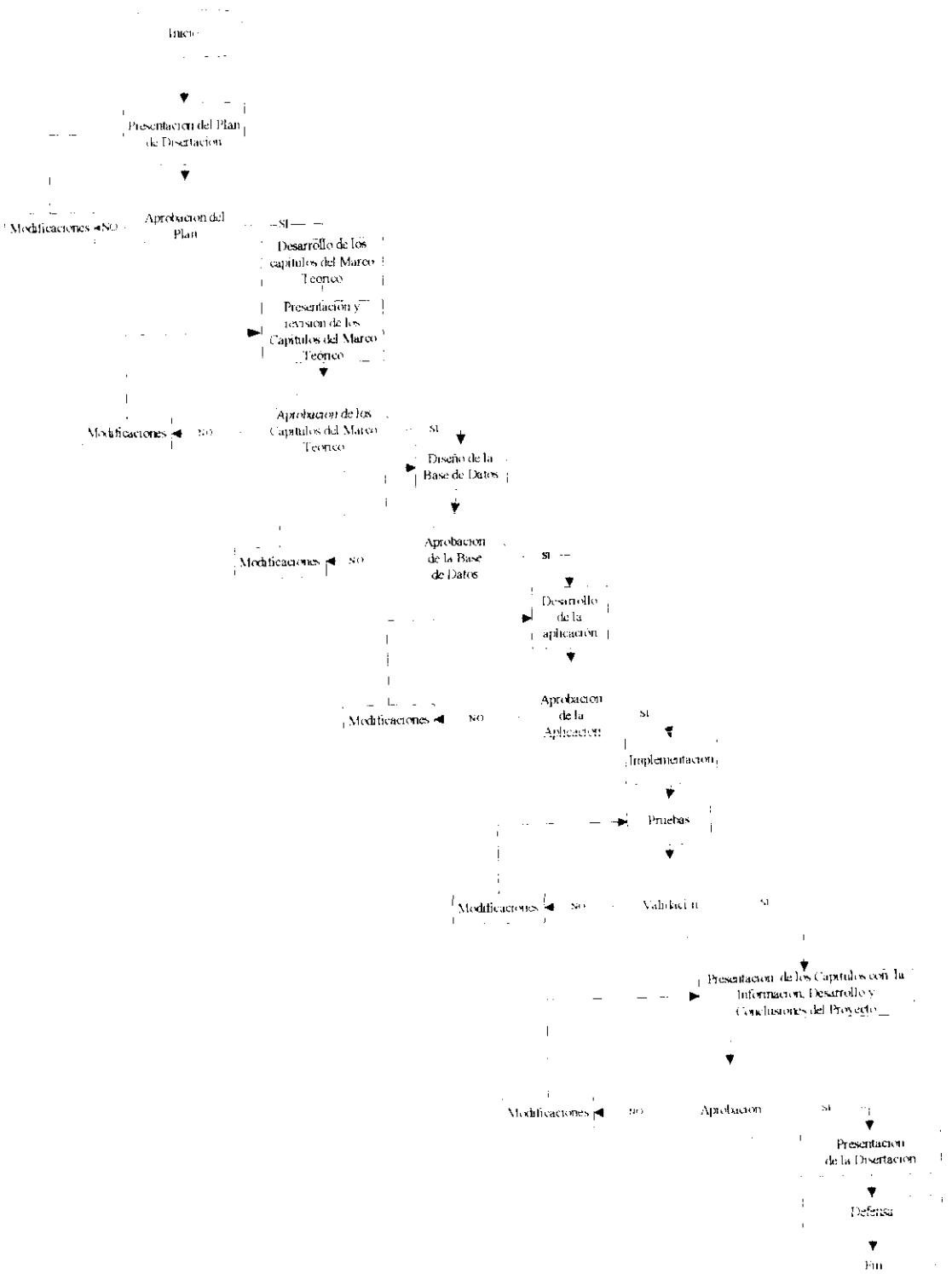
- El Paradigma que se utilizará en la disertación es la Racionalista por cuanto todo lo pensado puede ser transformado al mundo real, con la representación de programas.

El tipo de Investigación será Aplicada, por cuanto que todo lo investigado será puesto en acción, mediante diagramas de flujo y procedimientos.

Existen varios mecanismos que serán utilizados así como:

- Investigación Bibliográfica.- En la propuesta de la elaboración del marco teórico.
- Consultas con Expertos.- En la propuesta de la elaboración de documentación para la utilización de Visual Studio .Net se utilizará este mecanismo ya que se realizará consultas a personas que sepan acerca del manejo de herramientas .NET

1.5. PLAN OPERATIVO



CAPITULO II

MARCO TEÓRICO

2.1. Crear y mantener una Base de Datos

Una base de datos es un conjunto, colección o depósito de datos almacenados en un soporte informático de acceso directo.

Uno de los objetivos principales de una base de datos es proporcionar a los usuarios una visión abstracta de los datos. Es decir, el sistema oculta ciertos detalles relativos a la forma en que se almacenan y mantienen los datos.

2.1.1. Bases de Datos

Es una colección de archivos interrelacionados, son creados con DBMS. (Data Base Management System) un conjunto de programas que se encargan de manejar la creación y todos los accesos a las bases de datos. El objetivo de un sistema manejador base de datos es proporcionar un contorno que sea a la vez conveniente para ser utilizado al extraer, almacenar y manipular información de una base de datos. Todas las peticiones a la base de datos, se manejan centralizadam ente por medio de DBMS, ya que este funciona como interfase entre los usuarios y la base de datos.

El contenido de una base de datos engloba a la información concerniente de una organización, de tal manera que los datos estén disponibles para los usuarios, una finalidad de la base de datos es eliminar la redundancia o al menos minimizarla. Los tres componentes principales de un sistema de base de datos son el hardware, el software DBMS y los datos a manejar, así como el personal encargado del sistema.

Dada la importancia que tienen en el mundo real las interrelaciones entre los datos, es imprescindible que la base de datos sea capaz de almacenar éstas interrelaciones, al igual que hace con otros elementos, siendo ésta una diferencia esencial respecto a los ficheros donde no se almacenan las interrelaciones.

Existen tres niveles de abstracción en los que puede considerarse la base de datos: físico, conceptual y de visión.

- En el nivel físico se describe cómo se almacenan los datos en cuanto a detalles de estructuras de datos complejas del nivel más bajo.
- En el nivel conceptual, que es el siguiente nivel más alto de abstracción, se describe cuáles son los datos reales que están almacenados en la base de datos y qué relaciones existen entre los datos.
- El nivel de visión es más alto, en el cual se describe solo una parte de la base de datos y se presentan vistas diferentes de la misma base de datos a los usuarios.

El área de Base de datos permite crear nuevas Bases de datos crear/modificar/eliminar Tablas, actualizar campos de dichas Tablas. También permite definir Estilos de presentación/Reglas de validación/Formatos de Datos; todos estos para ser utilizados por cualquier campo de las Tablas de la Base de Datos.

El sistema de gestión de la base de datos (SGBD) es una aplicación que permite a los usuarios definir, crear y mantener la base de datos, y proporciona acceso controlado a la misma.

El SGBD es la aplicación que interacciona con los usuarios de los programas de aplicación y la base de datos. En general, un SGBD proporciona los siguientes servicios:

Permite la definición de la base de datos mediante el lenguaje de definición de datos. Este lenguaje permite especificar la estructura y el tipo de los datos, así como las restricciones sobre los datos. Todo esto se almacenará en la base de datos.

Permite la inserción, actualización, eliminación y consulta de datos mediante el lenguaje de manejo de datos. El hecho de disponer de un lenguaje para realizar consultas reduce el problema de los sistemas de ficheros, en los que el usuario tiene que trabajar con un conjunto fijo de consultas, o bien, dispone de un gran número de programas de aplicación costosos de gestionar.

Hay dos tipos de lenguajes de manejo de datos: los procedurales y los no procedurales. Estos dos tipos se distinguen por el modo en que acceden a los datos. Los lenguajes

procedurales manipulan la base de datos registro a registro, mientras que los no procedurales operan sobre conjuntos de registros. En los lenguajes procedurales se especifica qué operaciones se deben realizar para obtener los datos resultado, mientras que en los lenguajes no procedurales se especifica qué datos deben obtenerse sin decir cómo hacerlo

El Sistema Gestor de Bases de Datos (SGBD) es un conjunto de programas, procedimientos y lenguajes que proporcionan a los usuarios las herramientas necesarias para operar con una base de datos. Por tanto, el SGBD actúa como un intermediario entre los usuarios y los datos. Debe cumplir una serie de funciones como descripción de los datos, de manera que debe permitir definir los registros, sus campos, sus relaciones de autorización, etc. Debe manipular los datos permitiendo a los usuarios insertar, suprimir, modificar y consultar datos de la base de datos y por último, debe permitir usar la base de datos, dando un interfaz adecuado a cada tipo de usuario.

2.1.1.1. Crear una Base de Datos

Sirve para planificar el tipo de información que se quiere almacenar en la misma, teniendo en cuenta dos aspectos: la información disponible y la información que se requiere.

La planificación de la estructura de la base de datos, en particular de las tablas, es vital para la gestión efectiva de la misma.

Se crea una base de datos, para planificar el tipo de información que se quiere almacenar en la misma, esta nos permite reducir la redundancia de datos al mínimo y agilizar las operaciones de consulta y actualización. .

La base de datos consiste en la creación de las tablas que la componen, lo que a menudo significa definir un espacio de nombres separado para cada conjunto de tablas.

2.1.1.2. **Modificar y Eliminar una Base de Datos**

Después de crear una base de datos, es posible cambiar su definición original. Entre otros cambios, pueden realizarse los siguientes:

- Ampliar el espacio asignado en la base de datos para el registro de transacciones o para los datos.
- Reducir el espacio asignado en la base de datos para el registro de transacciones y los datos.
- Agregar o quitar archivos de datos y de registro de transacciones.
- Crear grupos de archivos.
- Cambiar el grupo de archivos predeterminado.
- Cambiar la configuración de la base de datos.
- Desconectar bases de datos.
- Adjuntar nuevas bases de datos y separar bases de datos no utilizadas.
- Cambiar el nombre de la base de datos.
- Cambiar el propietario de la base de datos.

A continuación se detalla los diferentes tipos de borrados y de modificaciones:

- **Operación restringida**: esto es, el borrado o la modificación de tuplas de la relación que contiene la clave primaria referenciada; sólo se permite si no existen tuplas con dicha clave en la relación que contiene la clave ajena. Esto nos llevaría, por ejemplo, si se parte de la relación profesor_alumno y si se requiere borrar o eliminar un profesor, se debe verificar previamente que este profesor no tenga una relación con algún alumno y si fuera este el caso la eliminación de tal profesor sería imposible o restringido.

- **Operación con transmisión en cascada:** esto es, el borrado o la modificación de tuplas de la relación que contiene la clave primaria referenciada lleva consigo el borrado o modificación en cascada de las tuplas de la relación que contienen la clave ajena. En nuestro ejemplo, cuando se modifique un campo en este caso el nombre del profesor_alumno este equivaldría a que el campo nombre del profesor en la tabla alumno debe ser modificada por todos y cada uno de los alumnos; esta operación se automatiza con transición en cascada.

- **Operación con puesta a nulos:** esto es, el borrado o la modificación de tuplas de la relación que contiene la clave primaria referenciada lleva consigo poner a nulos los valores de las claves ajenas de la relación que referencia. Si en la relación profesor_alumno, todos los alumnos que se encuentren relacionados a un profesor tendrá un valor nulo en el campo relacional de la tabla alumnos admite el valor de nulo.

2.1.2. Tablas

La tabla es un medio de almacenar datos que organiza la información dentro de una base de datos. Las tablas tienen una estructura predefinida y contienen datos que se ajustan a esta estructura.

Las tablas organizan la información en filas y columnas. Dentro de una tabla, una fila de datos se denomina registro.

2.1.2.1. Diseñar Tablas

El diseño de la estructura de una tabla consiste en una descripción de cada uno de los campos que componen el registro y los valores o datos que contendrá; cada uno de esos campos.

La tabla de una base de datos es una colección de información homogénea relacionada entre si. El principal aspecto a tener en cuenta durante el diseño de una tabla es determinar claramente los campos necesarios, definirlos en forma adecuada con un nombre, su tipo y longitud.

Los elementos de esta información se denominan registros, cada registro esta conformado por idéntico número y tipo de campos. Cada campo tiene asociado un nombre; por el que se identifica, y un tipo de dato, que especifica la información que podrá almacenar dicho campo.

Atributo 1	Atributo 2	.....	Atributo n	
XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	**Tupla 1**
XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	**Tupla 2**
XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	.
XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	.
XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	**Tupla n**

Tabla # 1. Campos de la tablas

Una Tabla puede tener 0 o más registros y la definición de un registro deberá contar con al menos un campo.

2.1.2.2. Crear y modificar una tabla

Desde el momento en que la estructura de la base de datos se da sustancialmente por la unión de las tablas que la componen, su actualización corresponde a la eliminación y modificación de tablas o al cambio de sus características.

2.1.2.2.1. Crear una tabla

Contienen la información necesaria para el manejo de un proceso, entonces se dirá que para almacenar los datos de las entidades y relaciones de una base de datos se utiliza tablas, que está formado por un conjunto de filas y columnas donde cada fila contiene la información de un individuo y cada columna los valores de un atributo particular de la entidad.

Las tablas organizan la información en filas y columnas. Dentro de una tabla, una fila de datos se denomina registro, mientras que las columnas se denominan campos.

Un registro representa un elemento determinado de datos, por ejemplo, la entrada de una persona en una libreta de direcciones.

Un campo representa una subdivisión de los datos de un registro. Un registro que representa una entrada de una libreta de direcciones puede estar formado por los campos nombre y apellido, dirección, ciudad, código postal y número de teléfono, etc.

La implicación directa de esta disposición es que el número de celdas que obtendríamos sería el resultado de:

$$Max\{F_i, i = 1...n\} \times \left(\sum_{j=1}^{j=n} C_j \right)$$

F_i representa el número de filas de la tabla i,

C_j representa el número de columnas de la tabla j

n es el número de tablas que se obtendrían mediante un análisis relacional apropiado.

La cantidad de información redundante sería totalmente inaceptable para una base de datos mediana. La información redundante no sólo implica mayor necesidad de almacenamiento masivo, sino también una ralentización de todas las operaciones con los datos.

Se ofrece una buena solución a este problema, que nos permite reducir la redundancia de datos al mínimo y agilizar las operaciones de consulta y actualización. Lo que deberíamos hacer es separar la información que se refiere a las tres entidades que tenemos: profesores, niveles y semestres en tres relaciones independientes, y después relacionarlas entre sí. Los recuadros indican relaciones base, mientras que las flechas indican las interrelaciones entre ellas. Se dice que estas interrelaciones, en realidad, no existen a nivel de la base de datos. Los nombres de algunos atributos sugieren el tipo de claves que hemos usado: claves subrogadas.

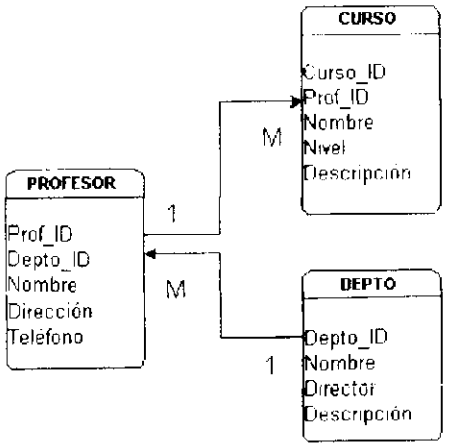


Tabla # 2. Base de datos

2.1.2.2.2. **Modificar una Tabla**

Las relaciones que existe no deben ser iguales, los campos de las tablas se modifica cuando los datos no están en la tabla o no se encuentran debidamente normalizados.

2.1.2.2.3. **Eliminar una tabla**

En determinadas ocasiones, necesitará eliminar una tabla por ejemplo, cuando desee implementar un nuevo diseño o liberar espacio en la base de datos. Cuando elimine una tabla, la definición de su estructura, sus datos, índices de texto, restricciones e índices se eliminan permanentemente de la base de datos y el espacio anteriormente utilizado para almacenar la tabla y sus índices queda disponible para otras tablas.

Si necesita eliminar tablas que están relacionadas mediante restricciones de clave externa (Foreign Key) y de no duplicados (Unique), o de clave principal (Primary Key), antes deberá eliminar las tablas con restricciones FOREIGN KEY. Si necesita eliminar una tabla a la que se hace referencia en una restricción FOREIGN KEY, pero no puede eliminar toda la tabla de clave externa, debe eliminar la restricción FOREIGN KEY.

2.1.3. Normalización

La normalización: se encarga de obtener los datos agrupados en distintas tablas siguiendo una serie de pasos, de tal manera que los datos obtenidos tienen una estructura óptima para su implementación, gestión y explotación desde distintas

aplicaciones futuras. Una de las ventajas principales que se obtiene al realizar la normalización es que la información no estará duplicada innecesariamente dentro de las estructuras: habrá mínima redundancia.

FORMAS NORMALES

Son las técnicas paras prevenir las anomalías en las tablas. Dependiendo de su estructura. Una tablas puede estar en primera forma normal. Segunda forma normal o cualquier otra. Relación entre las formas normales:

Primera forma normal (1FN):

"Una relación está en primera forma normal (1FN) si los valores para cada atributo de la relación son atómicos".

Esto quiere decir simplemente que cada atributo sólo puede pertenecer a un dominio y que tiene un valor único para cada fila.

La primera forma normal se definió para prohibir los atributos multivaluados, compuestos y sus combinaciones.

Cuando una relación no está en primera forma normal, se divide en otras relaciones, repartiendo sus atributos entre las resultantes. Normalmente la idea es eliminar el atributo que viola la 1ª FN de la relación original y colocarlo en una relación aparte junto con la clave primaria de la relación de partida.

Segunda forma normal (2FN):

"Una relación está en segunda a normal si está en la 1ª FN y todos los atributos no clave dependen de la clave completa y no sólo de una parte de esta".

Este paso sólo se aplica a relaciones que tienen claves compuestas, es decir, que están formadas por mas de un atributo. Si un esquema de relación no está en 2ªFN, se le puede normalizar a varias relaciones en 2ªFN en las que los atributos que dependen de una parte de la clave formarán una nueva relación que tendrá esa parte de la clave como clave primaria.

Tercera forma normal (3FN):

"Una relación está en tercera forma normal si todos los atributos de la relación dependen funcionalmente sólo de la clave, y no de ningún otro atributo".
Esto significa que en una relación en 3FN, para toda DF: X→Y, X es una clave.

Podrán observar que si una relación está en tercera forma normal, está también en segunda forma normal, sin embargo lo inverso no siempre es cierto.

LA CUARTA FORMA NOMAL ES DE BOYCE – CODD (FNBC)

Para definir la FNBC es conveniente introducir el término de Determinante. Un determinante se define como un atributo del cual depende funcionalmente algún otro atributo.

Una relación está en la FNBC si y solo si, todo determinante es una clave candidata. O sea, los únicos determinantes son las claves candidatas.

Los términos formales del modelo relacional a menudo son sustituidos por otros de uso más común, debido a que estos términos son demasiado abstractos para ser usados en la práctica. Así obtendrán las siguientes equivalencias:

Término relacional formal	**Equivalente informal**
Relación	Tabla
Tupla	Fila o registro
Cardinalidad	Número de filas o registros
Atributo	Columna o campo
Clave primaria	Identificador único
Dominio	Fondos de valores legales

2.1.4. Triggers

Un disparador (o trigger) es un procedimiento almacenado asociado a una tabla que se ejecuta al realizar una operación "básica" La operación básica que despierta al trigger es conocida como sentencia disparadora. La ejecución del disparador puede ser antes o después de llevar a cabo la sentencia disparadora. Es posible especificar condiciones adicionales para la ejecución del disparador. Dado que una sentencia disparadora puede afectar una o más filas de una tabla, es necesario especificar si se quiere que el disparador se ejecute para cada una de las filas afectadas o para el bloque en general.

Los disparadores son acciones que se "disparan" cuando se crea un registro (insert), se actualiza un registro (update) o se elimina un registro (delete).
Un database TRIGGER es una unidad de programa almacenado en la base de datos asociado con una tabla especifica de la base de datos

Al ejecutar el TRIGGER de la base de datos cuando la tabla asociada con este es afectada por una operación SQL, es decir los TRIGGERS son ejecutados de manera implicita.

Puede ser asociados un máximo de 12 TRIGGERS con una tabla de la base de datos.

Los TRIGGERS pueden ser empleados para:

- Verificar la modificación de datos en una tabla
- Transparencia de algún evento
- Derivación de valores en columnas de manera automática
- Implementación de Sistemas de seguridad Complejos
- Mantenimiento General de las Tablas

2.1.5. **Definición de Usuarios**

En un sistema de base de datos intervienen un número importante de usuarios, que podemos clasificar en tres grupos:

- Administrador de la base de datos (A.B.D.). Son los encargados de diseñar la estructura de la base de datos y los responsables de que el sistema funcione correctamente. El A.B.D. se encarga de autorizar el acceso a la base de datos, de coordinar y vigilar su utilización y de adquirir los recursos necesarios de software y hardware. El A.B.D. es el responsable cuando surgen problemas como violaciones de seguridad o una respuesta lenta del sistema. El A.B.D. tiene, entre otras, las -siguientes funciones:

  - Definición del esquema: Decidir el contenido de la base de datos, eligiendo cuales son los datos que interesa tener almacenados y organizarlos de la mejor forma posible, creando el esquema conceptual, que se escribirá mediante un lenguaje de definición de datos (DDL).

  - Definición de las estructuras de almacenamiento y método de acceso: Debe decidir sobre la forma en que se van a almacenar los datos sobre los soportes físicos en los que se grabará la base de datos y la correspondencia entre esta estructura de almacenamiento y el esquema conceptual.

  - Modificación del esquema y de la organización física si los requerimientos cambian.

  - Decidir los controles de autorizaciones para el acceso a los datos: Es el que concede diferentes tipos de autorizaciones al resto de los usuarios de la base de datos.

  - Especificar las restricciones de integridad: Debe definir los procedimientos de validación que habrán de ejecutarse cada vez que se

actualiza la base de datos. Estas restricciones son consultadas por el SGBD cada vez que se realiza una actualización de los datos.

- Programadores de aplicaciones, que se encargan de desarrollar las aplicaciones que manejan datos de la base de datos. Estas aplicaciones contendrán solicitudes de datos al S.G.B.D que luego serán procesados por los programas de la aplicación que tendrán como finalidad resolver problemas específicos de la empresa.

- Usuarios finales, que son personas que no tienen por que tener conocimientos informáticos y que pueden manipular los datos con la ayuda de las aplicaciones, o bien de lenguajes de consulta no procedimentales, tipo SQL, o bien, mediante herramientas basadas en sistemas de menús. Se distinguen tres tipos de usuarios finales:

- Usuarios especializados: Aquellos que son capaces de escribir ciertas aplicaciones para la BD, para su uso propio.

- Usuarios casuales: Aquellos que realizan consultas a través de un procesador de consultas. Esas consultas pueden ser creadas por ellos mismos o por otras personas.

- Usuarios ingenuos: Aquellos que solo acceden a través de aplicaciones previamente escritas por otros usuarios.

2.1.6. **Roles y privilegios**

El acceso a los objetos de la Base de Datos se realiza vía privilegios. Estos permiten que determinados comandos sean utilizados contra determinados objetos de la Base de Datos. Esto se especifica con el comando GRANT, conceder. Los privilegios se pueden agrupar formando lo que se conoce por roles. La utilización de los roles simplifica la administración de los privilegios cuando tenemos muchos usuarios. Los roles pueden ser protegidos con passwords, y pueden activarse y desactivarse dinámicamente, con lo que constituyen una capa más de seguridad en el sistema.

2.1.6.1. ROLES

Un rol puede tener garantizados una serie de privilegios tanto del sistema como sobre objetos, y a la vez puede tener garantizado otros roles..

Creación de Roles

Los Roles constituyen la forma más segura y rápida de asignar recursos a los grupos de usuarios. Es una tarea muy tediosa para cualquier DBA tener que asignar o revocar permisos a todos los usuarios, de a uno por uno, y es por eso que agrupando un conjunto de usuarios bajo las mismas características es posible manejar sus permisos como un grupo.

Para crear roles se utiliza la siguiente sintaxis:

CREATE ROLE nombre_rol NOT IDENTIFIED o IDENTIFIED BY password
Y para asignar el rol a un usuario o para comenzar a asignar/ quitar ciertos privilegios al rol se debe utilizar los comandos siguientes:

- GRANT: Otorga privilegios a un rol (o a un usuario cualquiera) o también asigna un rol a un usuario.
- REVOKE: Elimina privilegios otorgados previamente a un rol (o a un usuario).

Los roles o privilegios se pueden asignar varios al mismo usuario o grupo en una sola linea de comandos, siguiendo la sintaxis siguiente:

GRANT nombre_rol o nombre_privilegio [, nombre_rol o nombre_privilegio]
TO nombre_usuario o nombre_rol o PUBLIC [, nombre_usuario o nombre_rol]
[WITH ADMIN OPTION]

Esto hace que los usuarios anteriores posean el privilegio de SELECT. Cada uno de los usuarios que se incorporen a este rol, tendrá el mismo privilegio recién mencionado.

Si los privilegios se otorgan con la cláusula "with admin option" esto quiere decir que los usuarios que reciben los privilegios pueden a su vez otorgarlos a otros.

Eliminación de roles

DROP ROLE ROL_USUARIO

2.1.6.2. PRIVILEGIO

Un privilegio es un permiso dado a un usuario para que realice una operación. Estas operaciones pueden ser de dos tipos:

- Operación de sistema: necesita el permiso de sistema correspondiente.
- Operación sobre objeto: necesita el permiso sobre el objeto.

Un privilegio es: crear, modificar o borrar una tabla; consultar, insertar, borrar o modificar los datos de una tabla; consultar o crear una vista; crear usuarios o grupos; otorgar privilegios; etc.

Privilegios de sistema

Los privilegios de sistema son los que dan derecho a ejecutar o a realizar alguna orden sobre un objeto de algún tipo. Por ejemplo, disponer de privilegio para crear un tablespace.

Asignación de privilegios De sistema

GRANT { <privilegio> | <rol> }, . . .
TO { <usuario> | rol | PUBLIC }, . . .
| WITH ADMIN OPTION |
Donde:

- TO identica a los usuarios o roles que reciben los privilegios.
- La cláusula WITH ADMIN OPTION indica que el usuario que recibe los privilegios puede conceder estos a otros usuarios o roles.

Ejemplo: Cuando se crea un usuario en la base de datos, como mínimo deberán concederle privilegios para iniciar sesión en la misma.

Retirada de privilegios De sistema

REVOKE { <privilegio> | <rol> }. . . .

FROM { <usuario> | <rol> | PUBLIC }. . . .

3 GESTION DE TABLESPACES 5

Privilegios sobre objetos

Estos privilegios permiten acceder y realizar cambios en los datos de otros usuarios. Por ejemplo, poder consultar la tabla de otro usuario.

Sobre objetos

GRANT { <privilegio>, <privilegio> | ALL }

ON | <usuario>. |<objeto>

TO { <usuario> | rol | PUBLIC } . { <usuario> | rol | PUBLIC } ..

| WITH GRANT OPTIONS |

Ejemplo: A continuación, el usuario PROFESOR conceder privilegios de SELECT, INSERT y UPDATE al usuario USUARIO en la tabla DATOS:

GRANT SELECT, INSERT, UDPDATE ON DATOS TO USUARIO

Retirada Sobre objetos

REVOKE { <privilegio>, <privilegio> | ALL }

ON | <usuario>. |<objeto>

FROM { <usuario> | <rol> | PUBLIC }. . . .

Privilegios del Sistema

Los roles de sistema se utilizan para distribuir la disponibilidad de los comandos del sistema utilizados para gestionar la BD. Los privilegios más comunes están en la siguiente tabla. En ella se distinguen entre privilegios de manejo de objetos y de gestión de la BD. La palabra clave ANY significa que ese usuario tiene el privilegio para todos los esquemas en la BD. Hay que hacer notar que ANY y PUBLIC no son sinónimos.

Los privilegios se pueden agrupar en roles, para así satisfacer a distintos tipos de usuarios. En la instalación se crea un rol llamado OSOPER que sirve para los operarios de la máquina donde está la BD y permite realizar copias de seguridad en frió y en caliente. Los privilegios de OSOPER son STARTUP, SHUTDOWN, ALTER DATABASE OPEN/MOUNT, ALTER DATABASE BACKUP, ARCHIVE LOG, RECOVER y RESTRICTED SESSION.

Gestionando Privilegios

Los privilegios dan acceso a los usuarios a los datos que no poseen. Roles son grupos de privilegios que facilitan la administración de los privilegios. Pero los privilegios se pueden manejar de manera explícita en algunas circunstancias.

Los privilegios se crean vía el comando GRANT y son registrados en el diccionario de datos.

Haciendo un privilegio PUBLIC lo hace disponible a todos los usuarios de la BD.

Aunque los privilegios se puedan otorgar individualmente, no resulta razonable basar la gestión de los privilegios en su asignación individual. La gestión de los privilegios se facilita con la utilización de los roles

2.2. Tener Acceso y Cambiar Datos

Los datos existentes en una base de datos no son estáticas, sino que evolucionan en el tiempo. Existe, por tanto, la necesidad no sólo de añadir nuevos datos, sino de modificar los que están ya incluidos en las tablas de la base de datos

2.2.1. Agregar Datos

Es una entidad que contiene relaciones entre entidades. Consiste en construir un conjunto de entidades. La agregación se emplea para corregir algunos problemas que surgen cuando no es posible establecer una relación de otra relación.

Entidad.- cualquier objeto o evento acerca del cual alguien escoge o recolecta datos, puede ser un persona, cosa, etc

Existen casos donde será necesario agrupar dos o más conjuntos de entidades relacionados para conformar un solo conjunto lógico de entidades, a este proceso se le conoce como agregación. El objetivo primordial en la agregación será el establecer relaciones entre conjuntos de entidades agrupadas.

2.2.2. Cambiar Datos

Una vez que se han creado las tablas y se han agregado los datos, cambiar o actualizar los datos de las tablas se convierte en uno de los procesos diarios de mantenimiento de una base de datos.

tbl_alumno(cod_alumno, cedula,nombre-apellido)
tbl_profesor(cod_profesor,cedula,nombres)
tbl_nivel(cod_nivel,nombre)

2.2.3. Eliminar Datos

Si existen dos o más archivos con la misma información, los cambios que se hagan a éstos deberán hacerse a todas las copias del archivo de facturas.

tbl_alumno(cod_alumno, cedula,nombre-apellido)
tbl_profesor(cod_profesor,cedula,nombres)
tbl_nivel(cod_nivel,nombre)

2.3. Tecnologia .Net

La informática se inicio con programas monousuarios implantados en grandes ordenadores. Posteriormente estas primeras aplicaciones alcanzaron la capacidad de atender a diferentes usuarios. Pasaron los años y llego la arquitectura cliente-servidor, que gracias a este modelo de desarrollo, la aplicación se dividia en una parte que interaccionaba con el usuario y otra parte destinada al procesamiento de información.

En vistas de la evolución que está tomando el mercado de las comunicaciones en los últimos años, los ingenieros de Microsoft se han encerrado en sus laboratorios de Redmond durante más de una año para gestar su nueva criatura tecnológica a la que han llamado .NET.

Con ella pretenden cubrir el vacío existente en cuanto a soluciones unificadas para dispositivos móviles de última generación y tomar el puesto en Internet ocupado actualmente por Java.

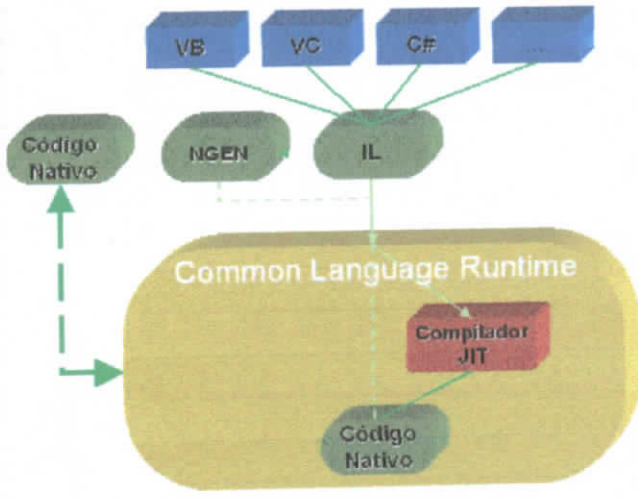


Grafico # 1. Componentes de Visual Studio.Net

Algunos investigadores de la industria de computación vaticinan que este cambio ser equivalente al que produjo la introducción de la PC, la interfase visual o al surgimiento mismo de Internet. Dispositivos móviles como celulares, TabletPC (PCs que parecen un cuaderno de notas pero tienen la capacidad de una computadora de escritorio), hasta televisores u otros dispositivos hogareños estarán conectados entre sí, con servidores y distintas aplicaciones. El elemento integrador será Internet. Estamos ahora en el inicio de la tercera generación de Internet. Con Visual Studio .NET y ASP.NET Web Matrix vamos a ser protagonistas del cambio.

Desde hace algún tiempo, Microsoft está llevando adelante una estrategia para construir una nueva tecnología tendiente a crear aplicaciones web distribuidas y que aprovechen al máximo las posibilidades que ofrece Internet.

Esta tecnología, que lleva el nombre de .NET, y que incluye un nuevo lenguaje denominado C#, una nueva versión de Visual Basic, con el nombre de Visual Basic.Net y otra serie de tecnologías, entre las que se encuentra: ASP.NET, que viene a reemplazar a las Active Server Pages (ASP), logrando el desarrollo de aplicaciones web más dinámicas, con un código más claro y limpio, por ende reusable,multiplataforma y definitivamente más simple, ya que el entorno ASP.NET permite la creación automática de alguna de las tarea más comunes para un creador web, cómo los formularios o la validación de los datos.

Los Ingenieros de Microsoft se han preocupado por brindarle a los desarrolladores un entorno de desarrollo que le permita disponer de una gran serie de herramientas y tecnologías tendientes a facilitar el desarrollo de aplicaciones web potentes y distribuidas, creando un ambiente multiplataforma, altamente deseado por todos los desarrolladores.

La nueva tecnología de Microsoft ofrece soluciones a los problemas de programación actuales, como son la administración del código o la programación por Internet. Para aprovechar al máximo las características del .net es necesario entender la arquitectura básica en la que esta implementada esta tecnología y así beneficiarse de todas las características que ofrece esta nueva plataforma

2.3.1. Microsoft .Net

La década de los 80's fue marcada por el surgimiento de la PC y de la interfase grafica. En la década de los 90's Internet permitió conectar computadoras en una escala global. En principio la conexión fue entre PCs y servidores por medio del explorador de Internet. A comienzos de este siglo es clara la necesidad de permitir a las computadoras conectadas a Internet comunicarse entre ellas. Desde entonces se va dando forma al nuevo modelo de computación distribuida llamado servicios Web basados en XML. El objetivo es permitir comunicarse entre sí a sistemas heterogéneos dentro y fuera de la

empresa. Esta comunicación es independiente del sistema operativo, lenguaje o modelo de programación. Para conseguir esto se desarrollaron estándares.

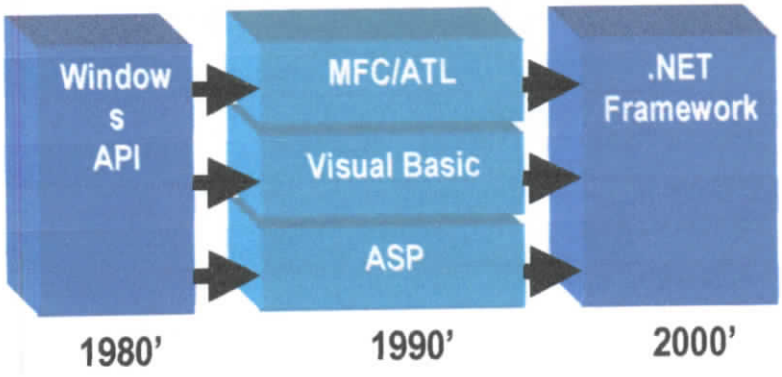


Grafico # 2 Tecnologia . Net

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.

Plataforma .NET?

Provee los cimientos para la nueva generación de software. Utiliza los Servicios Web como un medio para poder interoperar a distintas tecnologías. Permite conectar distintos sistemas operativos, dispositivos físicos, información y usuarios. Les da a los desarrolladores las herramientas y tecnologías para hacer rápidamente soluciones de negocios que involucran distintas aplicaciones, dispositivos físicos y organizaciones.

Servicios Web XML

Servicios existentes que implementan tareas predefinidas. Servicios hechos por el desarrollador.

Clientes

Dispositivos físicos que corren en sistemas operativos que integran e interactúan con otros elementos .NET.

Experiencia del usuario

Software tradicional que se integra con los servicios Web para presentar todo lo que el usuario necesita en una forma sencilla.

La idea central detrás de la plataforma .NET es la de servicio y construir, instalar, consumir, integrar o agregar estos servicios para que puedan ser accedidos mediante Internet. Esto es posible debido a que tenemos la infraestructura de comunicación global que es Internet cada vez mas rápida y a un costo cada vez menor y además, a la capacidad de los procesadores que continúa incrementándose año tras año. El usuario de Internet puede con un explorador de Internet no solamente acceder a contenido como texto, imágenes o sonido, también puede hacer uso de servicios Web. Estos son los bloques de construcción o componentes sobre los cuales se basa el modelo de computación distribuida en Internet. La plataforma .NET permite usar Internet y su capacidad de distribución para que los usuarios accedan desde cualquier dispositivo, en cualquier sistema operativo y lugar a servicios Web.

Los desarrolladores por su parte tienen la infraestructura y herramientas para crearv hacer uso de ellos en programas. Se aprovecha la capacidad de distribución a gran escala de Internet para acceder a servicios de software. También se aprovecha el incremento en la capacidad de procesamiento de los nuevos dispositivos móviles llamados "Smart Devices" (dispositivos inteligentes) para que el usuario haga uso de la funcionalidad que proveen los servicios Web con interfases cada vez más sencillas y naturales como la voz o la escritura.

- El Framework de .Net es una infraestructura sobre la que se reúne todo un conjunto de lenguajes y servicios que simplifican enormemente el desarrollo de aplicaciones. Mediante esta herramienta se ofrece un entorno de ejecución altamente distribuido, que permite crear aplicaciones robustas y escalables.

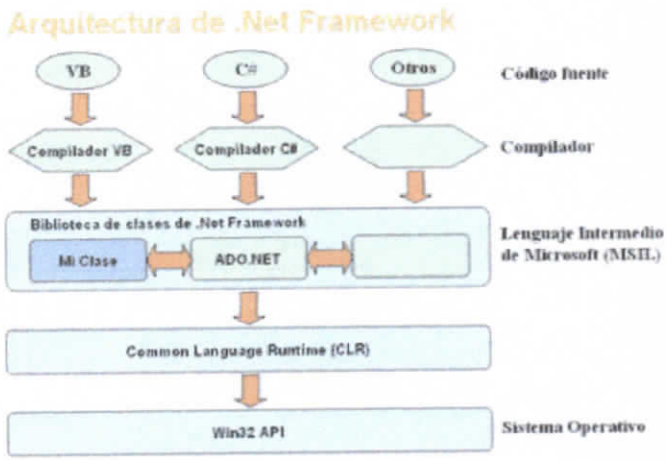


Gráfico # 4 Arquitectura del .Net Framework

Actualmente, el Framework de .Net es una plataforma no incluida en los diferentes sistemas operativos distribuidos por Microsoft, por lo que es necesaria su instalación previa a la ejecución de programas creados mediante .Net.

.Net Framework soporta múltiples lenguajes de programación y aunque cada lenguaje tiene sus características propias, es posible desarrollar cualquier tipo de aplicación con cualquiera de estos lenguajes. Existen más de 30 lenguajes adaptados a .Net, desde los más conocidos como C# (C Sharp), C++, Visual Basic hasta otros lenguajes menos conocidos como Perl o Cobol.

Web services.

Los servicios web son la revolución informática de la nueva generación de aplicaciones que trabajan colaborativamente en las cuales el software esta distribuido en diferentes servidores.

En este acercamiento se consiguió que cada una de las partes que constituían la aplicación pudiera residir en computadoras distintas. Con el paso del tiempo, la

computación aumento y llego la era de las aplicaciones distribuidas en las cuales los procesos se realizaban en diferentes unidades. De este paso surgió la tecnología Internet para solventar las problemáticas asociadas a fallo de aplicación centralizado.

2.3.1.1. Common Language runtime(CLR)

CLR es el verdadero núcleo del Framework de .Net, ya que es el entorno de ejecución en el que se cargan las aplicaciones desarrolladas en los distintos lenguajes, ampliando el conjunto de servicios que ofrece el sistema operativo estándar.

La herramienta de desarrollo compila el código fuente de cualquiera de los lenguajes soportados por .Net en un mismo código, denominado código intermedio (MSIL, Microsoft Intermediate Lenguaje). Para generar dicho código el compilador se basa en el Common Language Specification (CLS) que determina las reglas necesarias para crear código MSIL compatible con el CLR.

De esta forma, indistintamente de la herramienta de desarrollo utilizada y del lenguaje elegido, el código generado es siempre el mismo, ya que el MSIL es el único lenguaje que entiende directamente el CLR. Este código es transparente al desarrollo de la aplicación ya que lo genera automáticamente el compilador.
Sin embargo, el código generado en MSIL no es código máquina y por tanto no puede ejecutarse directamente. Se necesita un segundo paso en el que una herramienta denominada compilador JIT (Just-In-Time) genera el código máquina real que se ejecuta en la plataforma que tenga la computadora.

De esta forma se consigue con .Net cierta independencia de la plataforma, ya que cada plataforma puede tener su compilador JIT y crear su propio código máquina a partir del código MSIL.

La compilación JIT la realiza el CLR a medida que se invocan los métodos en el programa y, el código ejecutable obtenido, se almacena en la memoria caché de la computadora, siendo recompilado sólo cuando se produce algún cambio en el código fuente.

2.3.1.2. **Microsoft Intermediate Languaje (msil)**

Es un conjunto de instrucciones independiente de la CPU que se pueden convertir de forma eficaz en código nativo. MSIL incluye instrucciones para cargar, almacenar, inicializar y llamar a métodos en los objetos, así como instrucciones para operaciones lógicas y aritméticas, flujo de control, acceso directo a la memoria, control de excepciones y otras operaciones. Antes de ejecutar código, se debe convertir MSIL al código específico de la CPU, normalmente, mediante un compilador Just-In-Time (JIT).

Common Language Runtime proporciona uno o varios compiladores JIT para cada arquitectura de equipo compatible, por tanto, se puede compilar y ejecutar el mismo conjunto de MSIL en cualquier arquitectura compatible.

Cuando el compilador produce MSIL, también genera metadatos. Los metadatos describen los tipos que aparecen en el código, incluidas las definiciones de los tipos, las firmas de los miembros de tipos, los miembros a los que se hace referencia en el código y otros datos que el motor de tiempo de ejecución utiliza en tiempo de ejecución.

El lenguaje intermedio de Microsoft (MSIL) y los metadatos se incluyen en un archivo ejecutable portable (PE), que se basa y extiende el PE de Microsoft publicado y el formato Common Object File Format (COFF) utilizado tradicionalmente para contenido ejecutable. Este formato de archivo, que contiene código MSIL o código nativo así como metadatos, permite al sistema operativo reconocer imágenes de Common Language Runtime. La presencia de metadatos junto con el Lenguaje intermedio de Microsoft (MSIL) permite crear códigos autodescriptivos, con lo cual las bibliotecas de tipos y el Lenguaje de definición de interfaces (IDL) son innecesarios.

El motor de tiempo de ejecución localiza y extrae los metadatos del archivo cuando son necesarios durante la ejecución.

2.3.1.3. **Metadatos**

Los metadatos son un conjunto de datos organizados en forma de tablas que almacenan información.

Los metadatos son información binaria que describe un programa, almacenada en un archivo ejecutable portable (PE) de Common Language Runtime o en memoria. Cuando se compila el código en un archivo PE, los metadatos se insertan en una parte del archivo, mientras que el código se convierte en lenguaje intermedio de Microsoft (MSIL) y se inserta en otra parte del archivo. Cada tipo y miembro definido, o al que se hace referencia, en un módulo o ensamblado se describe en los metadatos. Cuando se ejecuta código, el motor de tiempo de ejecución carga los metadatos en la memoria y hace referencia a ellos para obtener información acerca de las clases, miembros, herencia, etc., del código.

En .NET se distinguen dos tipos de módulos de código compilado: ejecutables (extensión .exe) y librerías de enlace dinámico (extensión .dll generalmente) Ambos son ficheros que contienen definiciones de tipos de datos, y la diferencia entre ellos es que sólo los primeros disponen de un método especial que sirve de punto de entrada a partir del que es posible ejecutar el código que contienen haciendo una llamada desde la línea de comandos del sistema operativo. A ambos tipos de módulos se les suele llamar ejecutables portables (PE), ya que su código puede ejecutarse en cualquiera de los diferentes sistemas operativos de la familia Windows para los que existe alguna versión del CLR.

El contenido de un módulo no sólo MSIL, sino que también consta de otras dos áreas muy importantes: la cabecera de CLR y los metadatos:

La cabecera de CLR es un pequeño bloque de información que indica que se trata de un módulo gestionado e indica es la versión del CLR que necesita, cuál es su firma digital, cuál es su punto de entrada (si es un ejecutable), etc.

Los metadatos describen todos los tipos y miembros definidos en el código mediante un lenguaje neutro. Los metadatos almacenan la siguiente información:

Descripción del ensamblado
Identidad (nombre, versión, referencia cultural, clave pública).
Los tipos que se exportan.
Otros ensamblados de los que depende éste.

Permisos de seguridad que hay que ejecutar.

Descripción de los tipos.

Nombre, visibilidad, clase base e interfaces implementadas.

Miembros (métodos, campos, propiedades, eventos, tipos anidados).

Atributos

Elementos descriptivos adicionales que modifiquen los tipos y los miembros.

Ventajas de los metadatos

Los metadatos son la clave para un modelo de programación más sencillo, eliminando la necesidad de tener archivos de Lenguaje de definición de interfaz (IDL), archivos de encabezado o cualquier método externo de referencia a componentes. Los metadatos permiten que los lenguajes .NET se describan a sí mismos automáticamente usando un lenguaje de forma neutral, que no ven ni el programador ni el usuario

2.3.1.4. Ensamblados

Un ensamblado es una agrupación lógica de uno o más módulos o ficheros de recursos (ficheros .GIF, .HTML, etc.) que se engloban bajo un nombre común. Un programa puede acceder a información o código almacenados en un ensamblado sin tener porqué sabe cuál es el fichero en concreto donde se encuentran, por lo que los ensamblados nos permiten abstraernos de la ubicación física del código que ejecutemos o de los recursos que usemos.

Por ejemplo, podemos incluir todos los tipos de una aplicación en un mismo ensamblado pero colocando los más frecuentemente usados en un cierto módulo y los menos usados en otro, de modo que sólo se descarguen de Internet los últimos si es que se van a usar.

Todo ensamblado tiene metadatos con información sobre las características del ensamblado. Este manifiesta que puede almacenarse cualquiera de los módulos que formen el ensamblado o en uno específicamente creado para ello, caso éste último necesario cuando es un ensamblado satélite.

Hay dos tipos de ensamblados: ensamblados privados y ensamblados compartidos. Los privados se almacenan en el mismo directorio que la aplicación que los usa y sólo puede usarlos ésta, mientras que los compartidos se almacenan en un caché de ensamblado global (GAC) y pueden usarlos cualquiera que haya sido compilada referenciándolos.

2.3.2. Visual C#

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

Aunque es posible escribir código para la plataforma .NET en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella, por lo que programarla usando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes ya que C# carece de elementos heredados innecesarios en .NET. Por esta razón, se suele decir que C# es el lenguaje nativo de .NET

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

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

Características del 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# Alguna de las características aquí señaladas no son exactamente propias del lenguaje sino de la plataforma .NET en general. 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:

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

En lo referente a la encapsulación es importante señalar que aparte de los típicos modificadores public, private y protected, C# añade un cuarto modificador llamado internal, que puede combinarse con protected e indica que al elemento a cuya definición precede sólo puede accederse desde su mismo ensamblado.

C# se ha optado por hacer que todos los métodos sean por defecto sellados y que los redefinibles hayan demarcarse con el modificador **virtual** (como en C++), lo que permite evitar errores derivados de redefiniciones accidentales. Además, un efecto secundario de esto es que las llamadas a los métodos serán más eficientes por defecto al no tenerse que buscar en la tabla de funciones virtuales la implementación de los mismos a la que se ha de llamar. Otro efecto secundario es que permite que las llamadas a los métodos virtuales se puedan hacer más eficientemente al contribuir a que el tamaño de dicha tabla se reduzca.

- **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)

- **Gestión automática de memoria:** Como ya se comentó, todo lenguaje de .NET tiene a su disposición el recolector de basura del CLR. Esto tiene el efecto en el lenguaje de que no es necesario incluir instrucciones de destrucción de objetos. Sin embargo, dado que la destrucción de los objetos a través del recolector de basura es indeterminista y sólo se realiza cuando éste se active -ya sea por falta de memoria, finalización de la aplicación o solicitud explícita en el fuente-, C# también proporciona un mecanismo de liberación de recursos determinista a través de la instrucción **using**.

    - **Seguridad de tipos:** C# incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente, lo que

permite evita que se produzcan errores difíciles de detectar por acceso a memoria no perteneciente a ningún objeto y es especialmente necesario en un entorno gestionado por un recolector de basura.

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

- **Sistema de tipos unificado:** A diferencia de C++, en C# todos los tipos de datos que se definan siempre derivarán, aunque sea de manera implícita, de una clase base común llamada **System.Object**, por lo que dispondrán de todos los miembros definidos en ésta clase (es decir, serán "objetos")

- **Extensibilidad de tipos básicos:** C# permite definir, a través de **estructuras**, tipos de datos para los que se apliquen las mismas optimizaciones que para los tipos de datos básicos. Es decir, que se puedan almacenar directamente en pila (luego su creación, destrucción y acceso serán más rápidos) y se asignen por valor y no por referencia. Para conseguir que lo último no tenga efectos negativos al pasar estructuras como parámetros de métodos, se da la posibilidad de pasar referencias a pila a través del modificador de parámetro **ref**.

- **Extensibilidad de operadores:** Para facilitar la legibilidad del código y conseguir que los nuevos tipos de datos básicos que se definan a través de las estructuras estén al mismo nivel que los básicos predefinidos en el lenguaje, al igual que C++ y a diferencia de Java, C# permite redefinir el significado de la mayoría de los operadores -incluidos los de conversión, tanto para conversiones implícitas como explícitas- cuando se apliquen a diferentes tipos de objetos.

- **Extensibilidad de modificadores:** C# ofrece, a través del concepto de **atributos**, la posibilidad de añadir a los metadatos del módulo resultante de la compilación de cualquier fuente información adicional a la generada por el

compilador que luego podrá ser consultada en tiempo ejecución a través de la librería de reflexión de .NET . Esto, que más bien es una característica propia de la plataforma .NET y no de C#, puede usarse como un mecanismo para definir nuevos modificadores.

- **Versionable:** C# incluye una **política de versionado** que permite crear nuevas versiones de tipos sin temor a que la introducción de nuevos miembros provoquen errores difíciles de detectar en tipos hijos previamente desarrollados y ya extendidos con miembros de igual nombre a los recién introducidos.

    - Si una clase introduce un nuevo método cuyas redefiniciones deban seguir la regla de llamar a la versión de su padre en algún punto de su código, difícilmente seguirían esta regla miembros de su misma signatura definidos en clases hijas previamente a la definición del mismo en la clase padre; o si introduce un nuevo campo con el mismo nombre que algún método de una clase hija, la clase hija dejará de funcionar.

- **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.

- **Compatible:** Para facilitar la migración de programadores, C# no sólo mantiene una sintaxis muy similar a C, C++ o Java que permite incluir directamente en código escrito en C# fragmentos de código escrito en estos lenguajes, sino que el CLR también ofrece, a través de los llamados **Platform Invocation Services (PInvoke)**, la posibilidad de acceder a código nativo escrito como funciones sueltas no orientadas a objetos tales como las DLLs de la API Win32. Nótese que la capacidad de usar punteros en código inseguro permite que se pueda acceder con facilidad a este tipo de funciones, ya que éstas muchas veces esperan recibir o devuelven punteros.

También es posible acceder desde código escrito en C# a objetos COM. Para facilitar esto, el .NET Framework SDK incluye una herramientas llamadas **tlbimp** y **regasm** mediante las que es posible generar automáticamente clases proxy que permitan, respectivamente, usar objetos COM desde .NET como si de objetos .NET se tratase y registrar objetos .NET para su uso desde COM.

Finalmente, también se da la posibilidad de usar controles ActiveX desde código .NET y viceversa. Para lo primero se utiliza la utilidad **aximp**, mientras que para lo segundo se usa la ya mencionada **regasm**.

2.3.3. Visual J

VISUAL J# es un lenguaje de programación desarrollado por Microsoft muy apropiado para construir sistemas de información basados en red o mejor aun en Internet.

Desde el comienzo de J# en 2002, los miembros de la comunidad Java han solicitado la capacidad de migrar los subprogramas de Java a código administrado para poder ejecutarlos en el contexto de .NET Framework. Además de proporcionar más funciones y mayor seguridad que los subprogramas, esta tecnología resolverá los problemas de la máquina virtual Java de Microsoft. En respuesta a estas solicitudes de los clientes, Microsoft está desarrollando una tecnología conocida como J# Browser Controls.

Browser Controls permitirá a los desarrolladores migrar el código fuente del subprograma existente para ejecutarlo en .NET Framework sencillamente abriendo el código fuente y volviendo a compilar con unos cambios en el código mínimos, o ningunos. Una vez finalizada la conversión, los desarrolladores pueden incrustar J# Browser Control en una página Web mediante un procedimiento similar al que se utiliza actualmente para incrustar los subprogramas. Por otra parte, Browser Controls tiene acceso completo a .NET Framework e incluirá compatibilidad para el consumo de servicios Web XML.

J# también incluirá varias mejoras en el lenguaje para incrementar la interoperabilidad entre varios lenguajes y la compatibilidad con futuras características del sistema operativo Windows. En primer lugar, J# se convertirá en un Extensor CLS completo al agregar compatibilidad para la creación de tipos Enums y Value. En segundo lugar, J# agregará compatibilidad para las palabras clave assert y volatile para mejorar la flexibilidad del código en escenarios de código optimizado. Por último, J# proporcionará compatibilidad para el consumo de componentes genéricos para garantizar la coherencia entre lenguajes en futuras versiones de .NET Framework.

Con una sintaxis conocida y una funcionalidad orientada al objeto, Visual J# continúa siendo el medio más productivo para que los desarrolladores de lenguaje Java inicien la creación de aplicaciones con .NET Framework. Las mejoras de Visual J# Whidbey ampliarán la gama de capacidades disponibles para los estudiantes y profesores, así como para los desarrolladores de los actuales lenguajes Java y Visual J++.

Visual J # es una herramienta que los programadores del lenguaje Java pueden utilizar para construir usos y servicios para funcionar en el marco del NET.

Microsoft Visual J# .NET es una herramienta de desarrollo que pueden utilizar los programadores que conozcan la sintaxis del lenguaje Java para crear aplicaciones y servicios en .NET Framework. Integra la sintaxis Java en el intérprete de comandos de Visual Studio .NET. Visual J# .NET también es compatible con la mayoría de las funciones de Visual J++ 6.0, como las extensiones Microsoft. Visual J# .NET no es una herramienta para desarrollar aplicaciones pensadas para ejecutarse en una máquina virtual Java. Tanto las aplicaciones como los servicios creados con Visual J# .NET sólo se ejecutarán en .NET Framework. Visual J# .NET ha sido desarrollado de forma independiente por Microsoft.

Puesto que Visual J# .NET se integra con el entorno de desarrollo integrado (IDE) de Visual Studio .NET, los programadores de J# pueden utilizar diseñadores Visual Studio .NET para crear aplicaciones de servicios Web XML, Web Forms y Windows Forms.

Visual J# .NET Redistributable Package es el paquete redistribuible para Microsoft Visual J#. Este software ejecuta únicamente aplicaciones y servicios desarrollados con Visual J#; las aplicaciones de Java escritas con otras herramientas de desarrollo de Java no se ejecutarán con Microsoft Visual J# .NET Redistributable Package.

Visual J # apunta el tiempo de pasada del lenguaje común y se puede utilizar desarrollar usos del NET, incluyendo los servicios de XML, haciendo el uso completo del marco del NET. Visual J # los usos benefician del siguiente:

- Integración del lenguaje
- Seguridad realzada
- Versionable y ayuda del despliegue
- Eliminando errores y perfilando de la nota

La extensión de archivo de fuente en visual J # es jsl.

Visual J # incluye el siguiente:

- El visual J # recopilador, que compila fuentes del lenguaje Java de lengua intermedia de (MSIL).
- Un convertidor binario que transforma el bytecode del lenguaje Java en MSIL.
- Las bibliotecas independientemente desarrolladas de la clase diseñaron proporcionar la funcionalidad de la mayoría de las bibliotecas de la clase del nivel 1.1,4 del JDK y de muchas de las clases en el paquete del JDK **1,2** java.util
- Compatibilidad con las clases de la fundación de (WFC) y muchas del com.ms.

Sin embargo, Visual J #:

- No compila código de fuente de la Java-lengua al bytecode del lenguaje Java, es decir, archivos de los class.
- No apoya el desarrollo del applet o la capacidad de recibir applet en browsers o la capacidad de crear los usos que funcionarán en una máquina virtual de Java.

- No apoya el interfaz nativo de Java (JNI), el interfaz nativo (RNI), y la invocación alejada del método (RMI).

Microsoft visual J # no se utiliza para desarrollar los usos previstos para funcionar en una máquina virtual de Java. Los usos y los servicios construidos con visual J # funcionarán solamente en el marco del NET. Visual J # ha sido desarrollado independientemente por Microsoft.

J sharp es un recopilador que toma código usando el sintaxis de Java y lo compila en el seudo código del NET, que se conoce como el lenguaje intermedio de Microsoft (MSIL). Este código de MSIL entonces se liga a las varias bibliotecas y a funcionamiento en el tiempo de pasada del lenguaje común (o CLR), que es el ambiente de la ejecución del NET una máquina virtual relacionada con JVM de Java.

J # provee las bibliotecas que ponen muchos en ejecución de las clases y del APIs de Java como fueron definidos en Java 1,1,4. Específicamente, J # puede ligarse con éxito a las puestas en práctica del NET de las bibliotecas siguientes de la clase de Java:

java.io, java.lang, java.lang.reflect, java.math, java.net, java.security, java.sql, java.text, java.text.resources, java.util

ARQUITECTURA DE VISUAL J#

Visual J# incluye:

- Compilador de Visual J# (vjc.exe)
- Conversor binario de Visual J# (JbImp.exe)
- Un conjunto de bibliotecas de clases desarrolladas de manera independiente y diseñadas para proporcionar la funcionalidad de la mayoría de las bibliotecas de clases de JDK nivel 1.1.4 y muchas de las clases del paquete **java.util** de JDK 1.2
- Compatibilidad con las extensiones de Visual J++ 6.0

- Las bibliotecas de clases que se distribuyen con Visual J# son capas basadas en .NET Framework y Common Language Runtime, como se muestra la figura # .

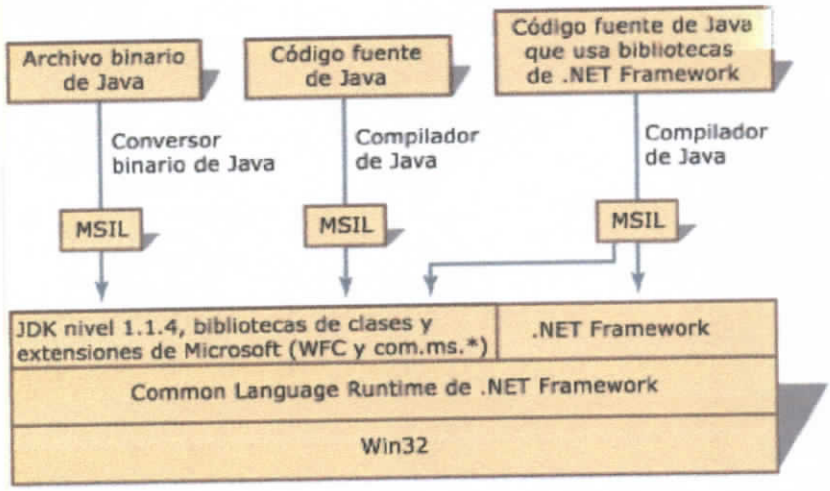


Gráfico # 5. Arquitectura de J#

Los archivos de código fuente de Java desarrollados con Visual J++ 6.0 se pueden compilar como un archivo ejecutable administrado .NET con el compilador de Visual J#. Determinadas aplicaciones disponibles sólo con formato de código de bytes se pueden convertir estáticamente en un archivo ejecutable administrado .NET con la herramienta Conversor binario de Visual J# .NET. Se pueden desarrollar aplicaciones de Visual J# nuevas para utilizar bibliotecas de clases de JDK nivel 1.1.4 compatibles, extensiones de Microsoft en Visual J++ 6.0 (por ejemplo, WFC y com.ms.*), así como .NET Framework.

Entorno de desarrollo integrado compartido

Microsoft Visual Studio .NET es el entorno de desarrollo integrado (IDE) compartido para todos los lenguajes que incorpora. Ha sido diseñado para ayudar a los desarrolladores a crear soluciones de forma rápida y con todas las herramientas disponibles con independencia del lenguaje que utilice en el IDE. Con Visual J# .NET, Java se une a los demás lenguajes compatibles con Visual Studio .NET. Para ver una introducción a las características de Visual Studio .NET,.

Diseñadores de Visual J# .NET

Diseñador de Web Forms

Los desarrolladores de hoy día precisan una forma intuitiva de crear páginas Web complejas. Visual Studio .NET incorpora el Diseñador de Web Forms compartido de Visual Studio, un modo gráfico de programar páginas HTML, páginas Active Server (ASP) y formularios Web Forms ASP.NET sin profundizar en código HTML o de secuencia de comandos.

Diseñador de Windows Forms

Windows Forms aporta un conjunto de clases conciso, orientado a objetos y ampliable que permite desarrollar rápidamente aplicaciones basadas en Windows. Con el Diseñador de Windows Forms, los desarrolladores pueden programar rápidamente soluciones para usar en aplicaciones basadas en Windows.

Con sólo agregar un nuevo formulario a un proyecto, el desarrollador tiene una base a partir de la cual puede crear rápidamente interfaces de usuario completas e intuitivas. Una vez que se han agregado formularios de Windows Forms a una solución de Visual Studio .NET, el desarrollador puede configurar propiedades de formularios, agregar controles del Cuadro de herramientas y escribir código para el formulario.

Diseñador de componentes

Del mismo modo que los diseñadores de formularios permiten crear aplicaciones cliente de forma rápida, Visual Studio .NET proporciona un modo de crear componentes de servidor de forma rápida y gráfica. El Diseñador de componentes aplica los conceptos de la programación basada en formularios RAD para crear objetos de nivel intermedio, una forma de crear objetos no visuales. En lugar de escribir grandes cantidades de código para el servidor, se pueden arrastrar los componentes del servidor a una superficie de diseño que se ejecutará en el servidor. Sólo tiene que hacer doble clic en un componente del servidor en el Diseñador de componentes y se abrirá el código de ese objeto.

Diseñador XML

El Diseñador XML proporciona herramientas intuitivas para trabajar con archivos XML y XSD (XML Schema Definition). Existen tres vistas en el diseñador: una para crear y editar esquemas XSD, otra para la edición estructurada de archivos de datos XML y otra para editar el código fuente XML.

2.3.4. Visual C++

En los últimos 10 años se han asentado diferentes técnicas de programación como la orientación a objetos, la programación basada en interfaces o los componentes. A pesar de ésto, los lenguajes de programación siempre han ido un paso por detrás de las mejores prácticas de programación del momento. Como resultado, los programadores tienden a depender de código específico, o simplemente a no usar las nuevas técnicas.

C++ soporta orientación a objetos, pero no tiene el concepto formal de interfaces. Entonces los programadores recurren a una mezcla entre clases abstractas bases e interfaces para simular programación basada en interfaces, mientras que utilizan modelos externos de componentes, como COM y CORBA para obtener los beneficios de la programación orientada a componentes.

Históricamente, dado que C++ fue creado por un desarrollador de ATT fue natural que esta empresa diera mucho apoyo a su difusión. Algo similar ocurrió con el lenguaje C y otros más (por ejemplo, SmallTalk que fue creado por XeroX). Pero un lenguaje de programación de amplia difusión, y libre, como lo es C++.

Windows es el entorno más popular de interfaz gráfico de usuario (GUI). Desde este punto de vista, Windows es un entorno multitarea basado en ventanas, que representan programas, y que permite ejecución concurrente.

Visual C++ es un entorno integrado de desarrollo que permite la programación orientada a objetos (POO) conjuntamente con el sistema de desarrollo SDK (también denominado API) de Windows. Al ser un entorno integrado Visual C++ incluye, entre otras, las siguientes herramientas de desarrollo:

- Editor de texto
- Compilador/Enlazador
- Depurador
- Visor de datos y dependencias (Browser)

Visual C++ incluye la librería de clases MFC (Microsoft Foundation Classes) que permite crear y gestionar de manera intuitiva componentes típicos de Windows. Esto es, la MFC es una implementación que utiliza el API encapsulando todas las estructuras y llamadas a funciones en objetos fáciles de utilizar. Basándose en la potencia de la MFC, Visual C++ se convierte en un generador de programas C++ para Windows.

Microsoft Visual C++ .NET proporciona el entorno de desarrollo dinámico para crear aplicaciones basadas en Microsoft Windows y Microsoft .NET, aplicaciones Web dinámicas y servicios Web XML mediante el lenguaje de programación C++. Visual C++ .NET incluye las bibliotecas estándar del sector Active Template Library (ATL) y Microsoft Foundation Class (MFC), extensiones de lenguaje avanzadas y características de entorno de desarrollo integrado (IDE) que permiten a los programadores editar y depurar código fuente de forma eficaz.

Proporciona a los programadores un lenguaje orientado a objetos probado para generar aplicaciones eficaces y sensibles al rendimiento. Con características avanzadas de plantilla, acceso de plataforma de bajo nivel y un compilador optimizado, Visual C++ .NET ofrece una funcionalidad superior para generar aplicaciones y componentes robustos.

El producto permite a los programadores generar una amplia variedad de soluciones, incluidas aplicaciones Web, aplicaciones de cliente inteligentes basadas en Microsoft Windows y soluciones para dispositivos móviles de cliente sencillos o inteligentes. C++ es el lenguaje de nivel de sistemas más popular del mundo y Visual C++ .NET ofrece a los programadores una herramienta universal con la que generar software.

El nuevo Visual C++ de Microsoft proporciona un entorno general, actualizado, a nivel de producción para el desarrollo de todas las aplicaciones para Windows 95/98 y

Windows NT. La versión 6 de Visual C++ de Microsoft abarca tres configuraciones diferentes: Ediciones Básicas, Profesionales y para Desarrollo.

El sistema de desarrollo

El compilador Visual C++ de 32 bits de Microsoft para Windows 95/98 y Windows NT incorpora nuevas herramientas de desarrollo de Windows completamente integradas y una interfaz visual. Por ejemplo, las capacidades de depuración del CodeView original de Microsoft son ahora directamente accesibles desde el depurador integrado del compilador como:

- El nuevo depurador integrado
- Los nuevos editores de recursos integrados
- El editor de cuadros de diálogo
- Los editores de imágenes
- El editor binario
- El editor de cadenas
- Herramientas adicionales

2.3.5. Visual Basic .NET

Con base en el sistema de tipos de .NET, Visual Basic.NET ofrece un número de mejoras importantes sobre versiones anteriores de Visual Basic. Las principales se relacionan con el soporte para las nociones de orientación a objetos como polimorfismo, herencia, y sobrecarga del operadores y métodos. Otras provienen de sus bases comunes con otros lenguajes proporcionados por CLS. Es posible que libremente pueda pasar tipos de datos y desde componentes desarrollados en otros lenguajes, y heredar desde clases base desarrolladas en otros lenguajes. Visual Basic.NET soporta ADO.NET para el acceso a datos, el modelo de programación Windows Forms, multiples hilos de ejecución, y nuevos y mejorados asistentes y diseñadores.

Los actuales desarrolladores en Visual Basic se beneficiarán de aprender las nuevas construcciones y características del lenguaje, permitiéndoles desarrollar aplicaciones

más avanzadas. Sin embargo, no tiene necesidad de actualizar todo su código, ya que el código Visual Basic 6 y el código Visual Basic.NET se ejecutará de lado a lado sin problemas. Si elige actualizar todo, los servicios de interoperabilidad COM ayudarán en gran medida con la transición por fases, permitiéndole utilizar sin problemas sus componentes existentes. Visual Basic.NET proporciona un asistente de actualización, el cual lo guía a través de los pasos del proceso de actualización y realiza los cambios de sintaxis de lenguaje necesarios cuando un proyecto Visual Basic es cargado.

Las principales nuevas características presentadas por Visual Basic.NET incluyen:

- Herencia. A través de la herencia de implementación, las clases desde una clase base existente. Visual Basic.NET (junto con todos los lenguajes .NET) únicamente soporta herencia simple, dando entender con esto que se puede heredar únicamente desde una sola clase base. Las nuevas clases pueden obtener la funcionalidad y comportamiento de la clase base, y prefirieren adaptar el comportamiento. Una clase derivada puede elegir proporcionar nuevos comportamientos, por ejemplo mediante la implementación de nuevos métodos. Para utilizar herencia de implementación, se declara la nueva clase y se utiliza la palabra reservada Inherits para expresar de qué clase base se deriva. Posteriormente utiliza la palabra reservada Overrides para modificar la funcionalidad de la clase base cuando corresponda. Observe que un método es capaz de ser sobreescrito en una clase derivada, debe ser marcado como Overridable en la clase base. Ésto se muestra en el siguiente código de ejemplo:

```
Public Class class1
  Overridable Function Name() As String
    ' Implementation goes here
  End Function
End Class
Public Class class2
  Inherits class1
  Overrides Function Name() As String
    ' Overriden Implementation goes here
  End Function
End Class
```

- Polimorfismo. Esta es la capacidad de tratar a los objetos de diferentes tipos de clase de la misma forma. Tradicionalmente, podría lograr esto en Visual Basic utilizando interfaces. Al tiempo que Visual Basic.NET sigue soportando el polimorfismo a través de interfaces, ahora también se puede lograr utilizando herencia.
- Constructores. Los constructores permiten que se cree un objeto y se inicie a un estado en una sola operación. Esto alivia la necesidad de crear un objeto, después llamar un método separado para inicializarlo. En Visual Basic.NET, los constructores de objetos le permiten pasar valores para un objeto mientras está siendo creado. Esto simplifica la codificación y ayuda a reducir errores cuando se trabaja con objetos.
- Inicializadores. Los inicializadores le permiten declarar una variable y proporcionarle un valor inicial en una sola línea de código. Esto permite lograr código más pequeño, más simple, y más mantenible.

- Manejo estructurado de errores. El CLR de .NET utiliza excepciones para manejar las condiciones de error de manera consistente, sin importar el lenguaje de programación que utilice. Las excepciones son respuestas a las condiciones de falla que pueden ocurrir mientras una aplicación se esté ejecutando. Pueden ser generadas por el Runtime de .NET (excepciones del sistema) o creadas programáticamente (excepciones de la aplicación). Las excepciones se pueden propagar más allá de los límites de los lenguajes. Por ejemplo, un componente escrito en C# podría generar una excepción con la que podría ser manejado subsecuentemente por un programa cliente desarrollado con Visual Basic.NET. Este código del sitio Web de Microsoft muestra la sintaxis Visual Basic.NET para usar el control estructurado de excepciones.

2.3.5.1.Proyectos Visual Basic

Visual Basic .Net es uno de los lenguajes de programación de alto nivel que pertenecen al .Net. con Visual Basic .Net se pueden escribir tanto programas convencional como para Internet. Las interfases podrán mostrar una interfaz grafica al usuario, o bien una interfaz de texto, como hacen las denominadas aplicaciones de consola.

El entorno .NET incluye un compilador de Visual Basic .Net que produce un código escrito en un lenguaje intermedio que será el que la maquina virtual ejecutara.

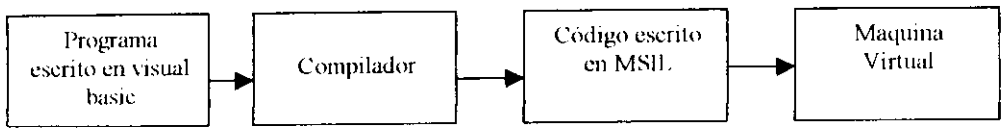


Grafico # 6 Proyecto en Visual Basic

Visual Basic sobre otros lenguajes de programación es que es independiente de la plataforma. El código producido por el compilador Visual Basic net puede transportar a cualquier plataforma .

En Visual Basic.Net lo que el usuario necesita para desarrollar un programa:

- Editar el programa
- Compilarlo
- Ejecutarlo
- Depurarlo

2.3.5.2.Programación Web

Los servicios Web XML permiten intercambiar datos en forma de mensajes XML entre sistemas heterogéneos. El acceso remoto a datos y lógica de aplicaciones es a modo de correspondencia imprecisa, sí lo es. Los anteriores, como DCOM, IIOP y Java/RMI, requieren una integración entre el cliente y el servidor y utilizan formatos de datos binarios específicos de la plataforma y la implementación. Los protocolos exigen una determinada tecnología de componentes o convención de llamada a objetos, los servicios Web XML no están sujetos. La única suposición existente entre el cliente y el servidor es que los destinatarios entenderán los mensajes que reciban. De otra forma, el cliente y el servidor aceptan un contrato, en este caso descrito utilizando WSDL y XSD, y luego se comunican mediante la generación de mensajes con arreglo al contrato a través de un determinado transporte como HTTP. Como resultado, los programas escritos en cualquier lenguaje que utilicen cualquier modelo de componentes y se ejecuten en cualquier sistema operativo podrán tener acceso a los servicios Web XML.

La flexibilidad que aporta el uso de un formato de texto como XML permite que el intercambio de mensajes evolucione con el tiempo en modo de correspondencia

imprecisa. Esta correspondencia imprecisa es obligatoria en entornos donde no es posible la actualización simultánea de todas las partes que intervienen en el intercambio de mensajes.

Los servicios Web XML se crean utilizando el entorno de aplicación ASP.NET, lo que permite a estos servicios Web XML tener acceso a numerosas funciones de .NET Framework, como autenticación, almacenamiento en caché y administración de estados. ASP.NET y .NET Framework constituyen la base de los servicios Web XML en código administrado.

Existen dos funciones fundamentales cuando se trabaja con los servicios Web XML:

- **Creación de un servicio Web XML.** Cuando se crea un servicio Web XML, se crea una aplicación que expone funcionalidad a clientes de servicios Web XML.
- **Acceso a un servicio Web XML.** Cuando se tiene acceso a un servicio Web XML, la aplicación de cliente busca, utiliza y hace referencia a la funcionalidad que contiene el servicio Web XML. El cliente de un servicio Web XML normalmente es una aplicación que puede enviar, recibir y procesar mensajes hacia y desde el servicio Web XML. En cambio, el único requisito es que el cliente debe ser capaz de, al menos, enviar mensajes al servicio Web XML. Esto incluye todas las aplicaciones que utilizan .NET Framework.

Los servicios Web XML pueden ser aplicaciones independientes o subcomponentes de una aplicación Web

2.3.5.3. El acceso a datos en la Web

Los servicios Web necesitan consultar o actualizar los datos persistidos ya sea en archivos simples, bases de datos relacionales o en cualquier otro tipo de almacén de datos. Para proporcionar el acceso a datos, los servicios del entorno incluyen una biblioteca de clase ADO.NET. Como el nombre lo indica, ADO.NET ha evolucionado a partir de los Objetos de Datos ActiveX (ADO).

ADO.NET está diseñado para proporcionar servicios de acceso a datos para aplicaciones y servicios escalables basados en Web . ADO.NET proporciona APIs de flujo de datos de alto desempeño para el acceso conectado estilo de cursor a datos, así como un modelo desconectado a datos más adecuado para retornar datos a las aplicaciones del cliente. Para la arquitectura ADO.NET, cualquier dato (sin importar cómo están actualmente almacenados) puede manipularse como XML o datos relacionales; lo que sea más importante para la aplicación en un momento determinado.

ADO.NET define clases para conectarse, emitir comandos y recuperar resultados de almacenes de datos. Los proveedores administrados de datos implementan estas clases. Los objetos connection y command de ADO.NET son similares a sus equivalentes en ADO, y una nueva clase denominada DataReader provee la capacidad de recuperar los resultados a través de una API de flujo de datos de alto desempeño. DataReader es equivalente funcionalmente a un Recordset de ADO de solo avance, sólo lectura; pero los DataReaders están diseñados para disminuir el número de objetos en memoria creados para evitar la recolección de basura y mejorar el desempeño. El entorno.NET incluye proveedores administrados de datos para Microsoft SQL Server y cualquier almacén de datos que puede accederse a través de OLE DB.

Una de las principales innovaciones de ADO.NET es la introducción de Dataset. Dataset es un caché de datos en memoria proporcionando una visión relacional de los datos. Consiste de una o más tablas junto con cualquier relación entre tablas y restricciones asociada. Los Datasets no saben nada acerca de la fuente de sus datos. Los Dataset se pueden crear y poblar programando o bien cargando datos desde el almacén de datos. Sin importar de donde provengan los datos, se manipula utilizando el mismo modelo de programación y utiliza la misma memoria caché.

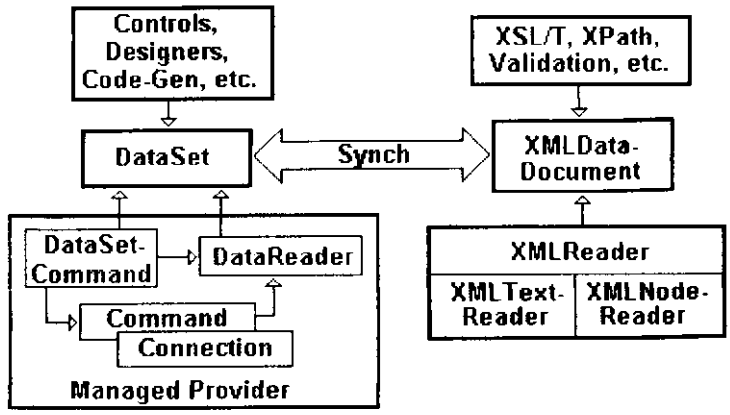


Grafico # 7 Arquitectura básica de ADO.NET

2.3.5.4.Servicios web XML

El XML proviene de un lenguaje que inventó IBM allá por los años 70. El lenguaje de IBM se llama GML (General Markup Language) y surgió por la necesidad que tenían en la empresa de almacenar grandes cantidades de información de temas diversos.

Por el año 89, para el ámbito de la red Internet, un usuario que había conocido el lenguaje de etiquetas (Markup) y los hiperenlaces creo un nuevo lenguaje llamado HTML, que fue utilizado para un nuevo servicio de Internet, la Web. Este lenguaje fue adoptado rápidamente por la comunidad y varias organizaciones comerciales crearon sus propios HTML, inventándose etiquetas. Desde el 96 hasta hoy una entidad llamada W3C ha tratado de poner orden en el HTML y establecer sus reglas y etiquetas para que sea un estándar. Sin embargo el HTML creció de una manera descontrolada y no cumplió todos los problemas que planteaba la sociedad global de Internet.

XML es una tecnología en realidad muy sencilla que tiene a su alrededor otras tecnologías que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores.

XML, con todas las tecnologías relacionadas, representa una manera distinta de hacer las cosas, más avanzada, cuya principal novedad consiste en permitir compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes. Así pues, el XML juega un papel importante, que tiende a la globalización y la compatibilidad entre los sistemas, ya que es la tecnología que permitirá compartir la

información de una manera segura, fiable, fácil. Además, XML permite al programador y los soportes dedicar sus esfuerzos a las tareas importantes cuando trabaja con los datos, ya que algunas tareas como la validación de estos o el recorrido de las estructuras a cargo del lenguaje y está especificado por el estándar.

Los servicios web son la revolución informática de la nueva generación de aplicaciones que trabajan colaborativamente en las cuales el software esta distribuido en diferentes servidores.

La informática se inicio con programas monousuarios implantados en grandes ordenadores. Posteriormente estas primeras aplicaciones alcanzaron la capacidad de atender a diferentes usuarios. Pasaron los años y llego la arquitectura cliente-servidor, que gracias a este modelo de desarrollo, la aplicación se dividía en una parte que interaccionaba con el usuario y otra parte destinada al procesamiento de información. En este acercamiento se consiguió que cada una de las partes que constituian la aplicación pudiera residir en computadoras distintas. Con el paso del tiempo, la computación aumento y llego la era de las aplicaciones distribuidas en las cuales los procesos se realizaban en diferentes unidades. De este paso surgió la tecnología Internet para solventar las problemáticas asociadas a fallo de aplicación centralizado. Véase documentación acerca de los orígenes de Internet

Un Web Service es un componente de software que se comunica con otras aplicaciones codificando los mensaje en XML y enviando estos mensaje a través de protocolos estándares de Internet tales como el Hypertext Transfer Protocol (HTTP). Un Web Service es similar a un sitio web que no cuenta con un interfaz de usuario y que da servicio a las aplicaciones en vez de a las personas. Un Web Service, en vez de obtener solicitudes desde el navegador y retornar paginas web como respuesta, lo que hace es recibir solicitudes a través de un mensaje formateado en XML desde una aplicación, realiza una tarea y devuelve un mensaje de respuesta también formateado en XML.

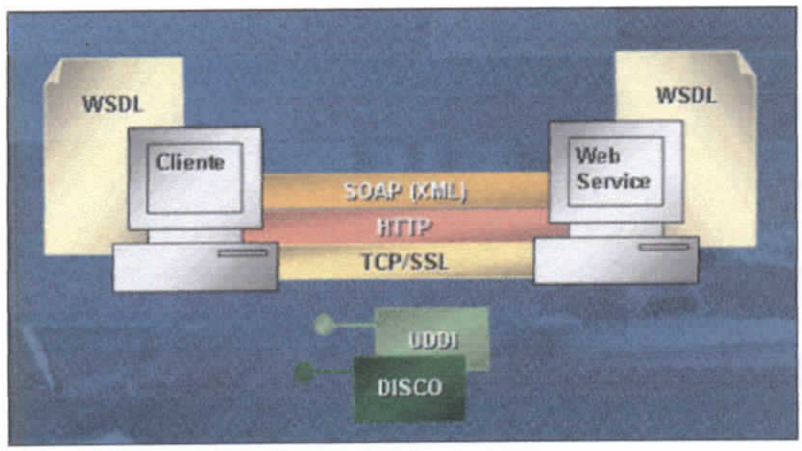


Gráfico # 8 Esquema básico de un Web Service

Los servicios XML Web Services son los elementos fundamentales en la evolución hacia la computación distribuida a través de Internet. Se están convirtiendo en la plataforma de integración de aplicaciones gracias a los estándares abiertos y al énfasis en la comunicación y colaboración entre personas y aplicaciones. Las aplicaciones se crean utilizando los servicios XML Web Services múltiples de origen distinto que funcionan conjuntamente, sin importar su ubicación o la forma en que se implementaron.

Existen tantas definiciones de los servicios XML Web Services como empresas que los diseñan. Sin embargo la idea general es:

- Los servicios XML Web Services ofrecen funciones muy útiles a usuarios del medio Web ya que emplean un protocolo Web estándar que, en casi todos los casos, es SOAP.
- Los servicios XML Web Services permiten describir sus interfaces con suficiente detalle para que el usuario diseñe una aplicación cliente que permita comunicarse con ellas. Esta descripción se proporciona normalmente en un documento XML denominado WSDL (lenguaje de descripción de servicios Web).
- Los servicios XML Web Services se registran para que los futuros usuarios los encuentren fácilmente. Este registro se realiza a través de UDDI (descripción, descubrimiento e integración universales).

Una de las principales ventajas de la arquitectura de los servicios XML Web Services es que permite que los programas escritos en lenguajes y plataformas diferentes puedan establecer comunicación entre sí de forma estándar.

Simple Object Access Protocol (SOAP): Protocolo de intercambio de información estructurada basado en XML. Entre otras cosas, SOAP puede representar llamadas a procedimientos remotos (RPC) y sus respuestas, proveyendo un mecanismo para intercambiar información y servicios entre aplicaciones. Funciona normalmente sobre HTTP así que trabaja con los firewalls, los proxies y otros obstáculos existentes. También puede utilizar la autenticación y el formato de compresión de HTTP y soporta SSL para encriptación y seguridad. Por todo esto es utilizado por los Servicios Web para la comunicación entre el proveedor de servicio y sus clientes.

WSDL: Documento XML que describe completamente el formato de los mensajes intercambiados. Se puede consultar libremente por los clientes y representa el "contrato" de servicio.

Web Service Discovery (DISCO): Es el proceso de localización y descubrimiento de las descripciones de servicios Web. A través de este proceso los clientes descubren la existencia del servicio, la funcionalidad ofrecida y como interactuar con él.

UDDI Business Registry: Los nodos UDDI permiten que las organizaciones publiquen información relativa a los servicios que ofrecen. Constituyen el punto de entrada a los servicios Web, proporcionando un mecanismo sencillo de localización de los servicios disponibles. Como parte de la especificación UDDI, se dispone de una API que permite interactuar programáticamente con el registro.

Un servicio Web XML es un componente de software o una aplicación que expone sus funciones programáticamente a través de Internet o la intranet utilizando los protocolos estándar de Internet como son Simple Object Access Protocol (SOAP) para comunicación entre programas, y XML para representación de datos. Puede ser de ayuda pensarlo como programación componentizada a través del Web.

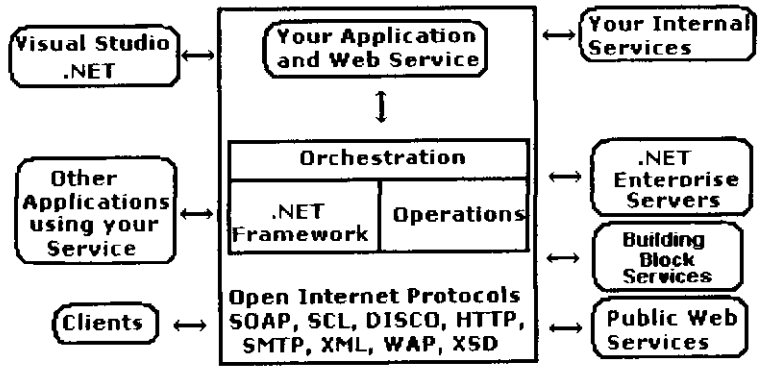


Grafico # 9 Modelo de Servicios Web

Integra los servicios Web externos en sus aplicaciones enrutando las solicitudes a través de Internet a un servicio que reside en un sistema remoto. Por ejemplo, un servicio como Microsoft Passport podría permitirle proporcionar autenticación para una aplicación. Mediante la programación del servicio Passport, puede aprovechar la infraestructura de Passport y confiar en ella para mantener la base de datos de los usuarios asegurando que sea operada de manera eficaz, respaldada regularmente, etc.

XML

XML es un lenguaje de metamarcado que ofrece un formato para la descripción de datos estructurados. Esto facilita unas declaraciones de contenido más precisas y unos resultados de búsquedas más significativos en varias plataformas. Además, XML habilitará una nueva generación de aplicaciones para ver y manipular datos basadas en el Web.

Ventajas de XML

XML aporta mucha potencia y flexibilidad a las aplicaciones basadas en el Web, proporcionando numerosas ventajas a los programadores y usuarios:

- Búsquedas con más significado
- Programación de aplicaciones Web flexibles
    - Integración de datos procedentes de fuentes dispares
    - Computación y manipulación locales de los datos
    - Varias vistas de los datos
    - Actualizaciones granulares

CAPITULO III

3. DESARROLLO

3.1. Desarrollo de la aplicación

Para desarrollar la aplicación SAGA_PUCESA(Sistema de Apoyo para la Gestión Academica de la PUCESA) del ingreso de notas y consultas de la mismas se ha realizado en la base de datos Anywhere Sql 5.0 con Visual Basic.Net .

La aplicación se desarrollo en Visual Basic.Net con AnyWhere Sql 5.0 dentro de la cual se consultara las notas de los alumnos, los profesores podrán ingresarán las notas como la asistencia de los alumnos esto nos permitirá ver de una manera rápida y eficaz ya que hasta la actualidad se lo realiza manualmente y se publica en las carteleras existentes en la Universidad.

3.1.1. Análisis del Diseño

Es un conjunto o disposición de procedimientos o programas relacionados de manera que juntos forman una sola unidad. Un conjunto de hechos, principios y reglas clasificadas y dispuestas de manera ordenada mostrando un plan lógico en la unión de las partes. Un método, plan o procedimiento de clasificación para hacer algo. También es un conjunto o arreglo de elementos para realizar un objetivo predefinido en el procesamiento de la Información. Esto se lleva a cabo teniendo en cuenta ciertos principios:

Debe presentarse y entenderse el dominio de la información de un problema.
Defina las funciones que debe realizar el Software.
Represente el comportamiento del software a consecuencias de acontecimientos externos.
Divida en forma jerárquica los modelos que representan la información, funciones y comportamiento.
El proceso debe partir desde la información esencial hasta el detalle de la Implementación.

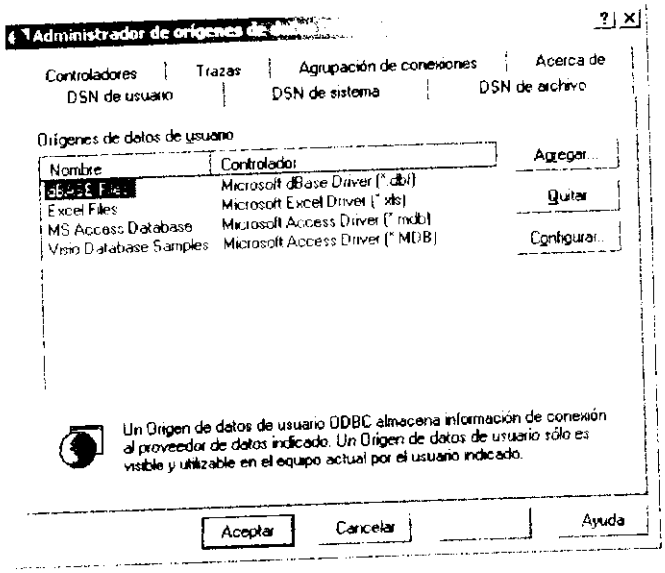
3.1.2. Análisis de la base de datos

La base de datos que se utiliza para el desarrollo del sistema es tomada de una base de datos ya existente desarrollado en la Disertación titulada "Desarrollo e Implementación del Sistema Académico Aplicado a la Escuela de Ingeniería de Sistemas de la PUCESA".

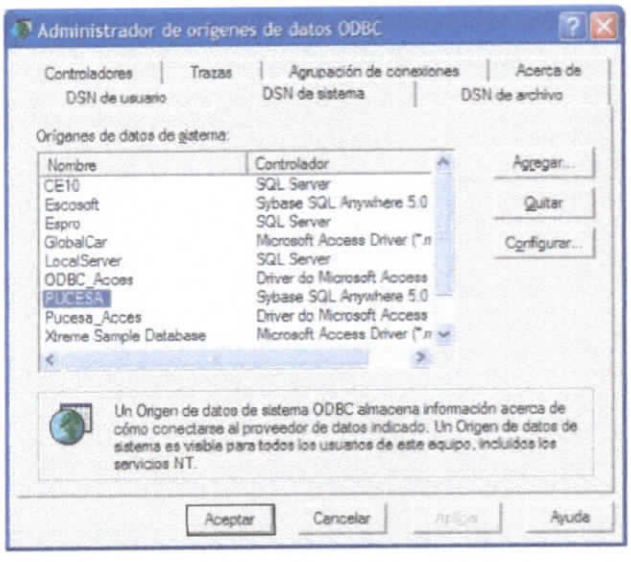
3.1.3. Implementación de la base de datos

CREACIÓN DEL ODBC

El ODBC es un programa que permite realizar una conexión entre el sistema Operativo de la computadora. En el panel de control del computador se ingresa en la opción Orígenes de datos (ODBC) dentro del icono Herramientas Administrativas. Se desplegara una ventana donde en la pestaña DSN (Data Source Name) del sistema y seleccione la opción agregar.



Luego se selecciona un controlador para el origen de la Base de datos. En nuestro caso es el controlador de My SQL anywhere 5.0.



Luego se levantara una base de datos creada

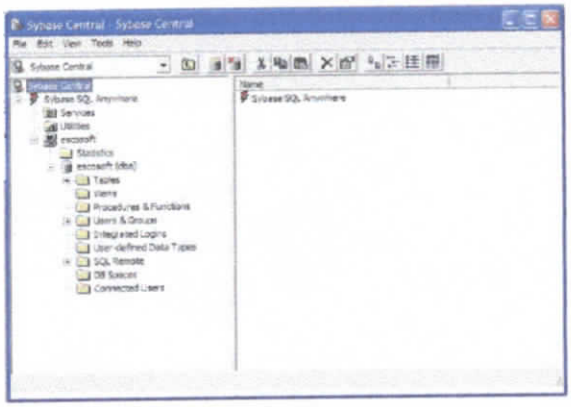


My SQL Anywhere 5.0

En esta ventana es donde el nombre de usuario la clave y el nombre de la base de datos.



En esta sección nos permite ver la base de datos ya creada en My SQL Anywhere 5.0



3.1.4. Diseño de Interfaz

El Diseño de Interfaz se define el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, un proceso o un Sistema, con suficientes detalles como para permitir su interpretación y realización fisica.

El sistema desarrollado tiene una interfaz de fácil manejo, colores amigables al usuario le permite tener un ambiente agradable de trabajo dentro de los cuales esta destinado para poder consultar sus notas.

A continuación en el siguiente diagrama se detalla el sistema:

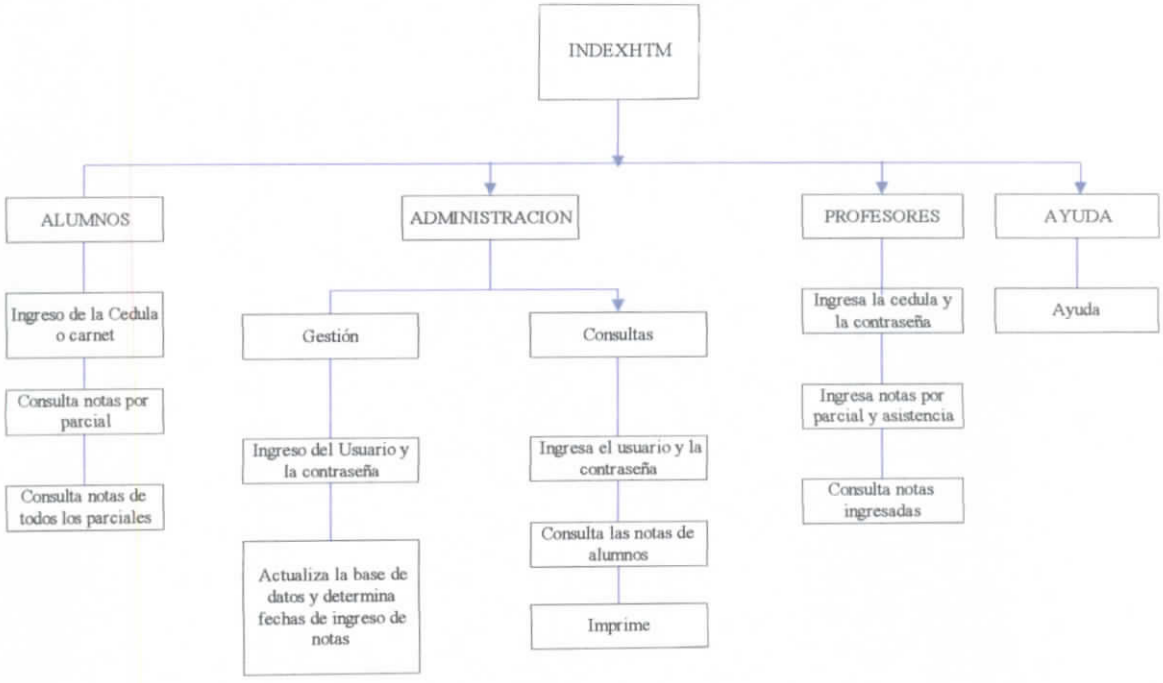


DIAGRAMA DE ITERACIÓN

En el diagrama se detalla la forma como esta conectada la base de datos con cada uno de los actores como son Profesor, Administrador, Alumnos:

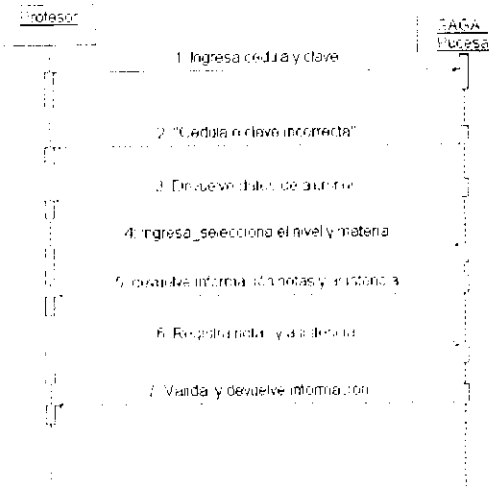


DIAGRAMA DE SECUENCIA

Un diagrama de secuencia representa la interacción entre diferentes objetos en el sistema, los diferentes objetos del diagrama de secuencia interactúan entre ellos a través del paso de "mensajes"

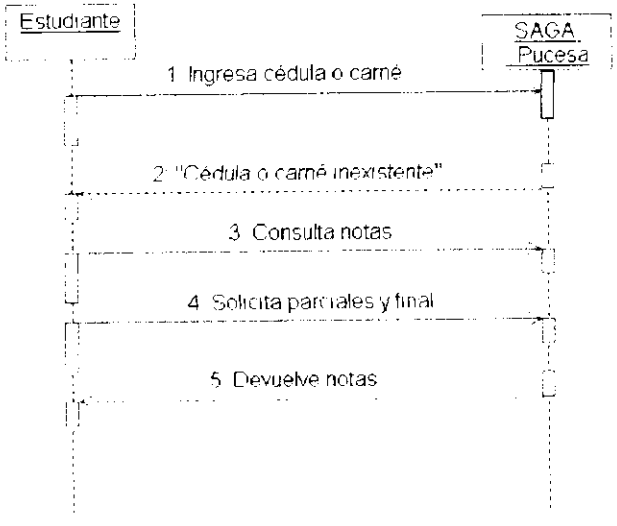
- **DIAGRAMA DEL PROFESOR**

  Es este diagrama se explica la forma como esta conectado los datos y de que manera interactúan los datos del profesor con la base de datos



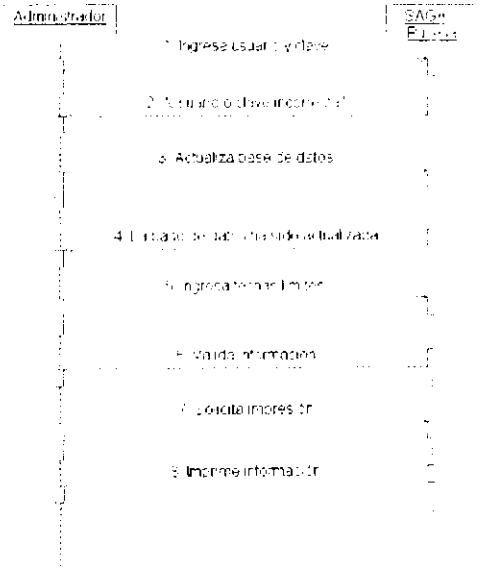
- **DIAGRAMA ESTUDIANTE**

  En esta sección se representa la opción de alumnos con la iteración del sistema en la cual esta demostrando como devuelve datos del sistema.



- **DIAGRAMA DEL ADMINISTRADOR**

  En el diagrama que tenemos continuación se detalla cada uno de los procesos que debe realizar el administrador en la base de datos.



MODELO ENTIDAD RELACIÓN

Es uno de los modelos lógicos basados en objetos y por lo tanto se enfoca primordialmente a los niveles conceptual y de visión. Una de las características de este modelo es que permite representar con claridad las limitantes de los datos. El modelo Entidad-Relación es en esencia una herramienta para representar el mundo real por medio de simbologías y expresiones determinadas.

Cuando se utiliza una base de datos para gestionar información, se está plasmando una parte del mundo real en una serie de tablas, registros y campos ubicados en un ordenador; creándose un modelo parcial de la realidad. Antes de crear físicamente estas tablas en el ordenador se debe realizar un modelo de datos.

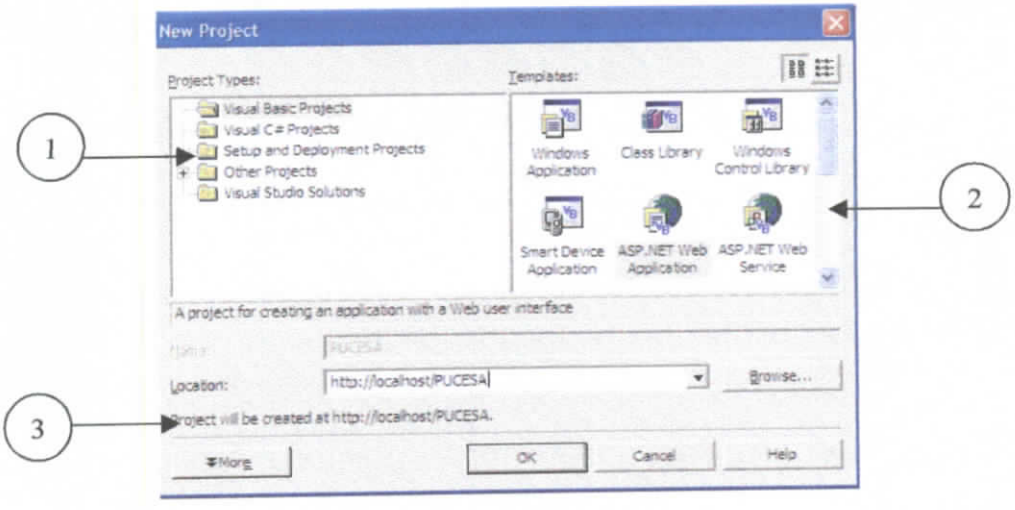
A continuación se detalla el modelo entidad relación de la base de datos SAGA_PUCESA.

3.1.5. Desarrollo de la aplicación Web en Visual Studio.Net

ENTORNO DE VISUAL STUDIO.NET

En esta pantalla se puede ver algunos de los componentes que tiene Visual Studio.Net Componentes de la ventana:

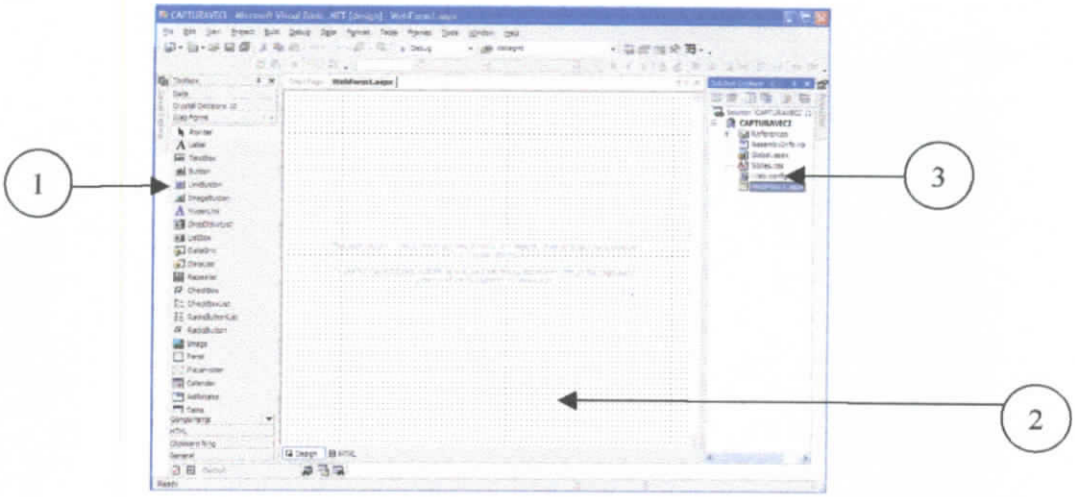
1. Tipos de proyectos
2. Plantillas de diseño de cada uno de los proyectos
3. Localización de base de datos



ENTORNO DE VISUAL BASIC.NET

En esta pantalla se presenta el diseño de interfaz de visual Basic.Net se detalla a continuación los siguientes componentes:

1. Barra de herramientas
2. Entorno de diseño
3. Explorador
    - Referencia
    - Información assembler de vb.
    - Estilos
    - Web form. Aspx



3.1.6. Implementación de la aplicación

Es el proceso instalar equipos o Software nuevo, como resultado de un análisis y diseño previo como resultado de la sustitución o mejoramiento de la forma de llevar a cabo un proceso automatizado.

Al Implantar un Sistema de Información lo primero que debemos hacer es asegurarnos que el Sistema sea operacional o sea que funcione de acuerdo a los requerimientos del análisis y permitir que los usuarios puedan operarlo.

Existen varios enfoques de Implementación:

- Asignar responsabilidad a los grupos.
- Entrenamiento de los usuarios. .

En la preparación de la Implantación, aunque el Sistema este bien diseñado y desarrollado correctamente su éxito dependerá de su implantación y ejecución por lo que es importante capacitar al usuario con respecto a su uso y mantenimiento.

Capacitación de Usuarios del Sistema:

Es enseñar a los usuarios que se relacionan u operan en un proceso de implantación.

Descripción de cada uno de los formularios del sistema

Nombre del archivo	Detalle
Ingreso.Aspx	Menú de acceso al sistema
IngresoAl.Aspx	Permite el ingreso de alumnos al sistema
Consultaralumno:Aspx	Permite consultar las notas al alumno por parcial
Consultafinal.Aspx	Consulta notas de los alumnos de todos los parciales
Ingresoad.Aspx	Ingreso del administrador al sistema
Administración.Aspx	Ingreso para actualizar la base de datos y la fecha al sistema
Consultaadministrador.Aspx	Consultas las notas de los alumnos el administrador o secretaria
DBPUCESA.Aspx	Actualiza la base de datos
ingresoP.Aspx	Ingreso del profesor al sistema
Notas.Aspx	Ingresa el Profesor las notas y la asistencia al sistema
ResumenGeneral:Aspx	Consulta general de las notas ingresadas al sistema
Principal.Html	Ayuda del sistema

DICCIONARIO DE DATOS

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

Contenido de un registro del diccionario

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

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

Descripción: Cada entrada en el diccionario consiste de un conjunto de detalles que describen los datos utilizados o producidos por el sistema.

Un nombre: para distinguir un dato de otro.

Longitud: porque es de importancia de saber la cantidad de espacio necesario para cada dato.

Valores de los datos: porque en algunos procesos solo son permitidos valores muy específicos para los datos. Si los valores de los datos están restringidos a un intervalo especifico, esto debe estar en la entrada del diccionario.

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

DICCIONARIO DE DATOS

Nombre de la Tabla	Definición	Campo	Tipo de Dato	Descripción	PK	FK
ma_accesos_menu	Permite registrar los accesos que tienen los usuarios al menu principal del sistema	maus_clave	VARCHAR(10)	Es la clave del usuario al acceso al menú	Yes	Yes
		maam_opcion_menu	VARCHAR(200)	El nombre de la opcion del menu	Yes	No
		maam_editar	VARCHAR(1)	s si puede editar y n si no puede editar	No	No
		maam_crear	VARCHAR(1)	S si puede editar y N si no puede editar	No	No
		maam_borrar	VARCHAR(1)	S si puede editar y N si no puede editar	No	No
		maam_imprimir	VARCHAR(1)	S si puede editar y n si no puede editar	No	No
		maam_opc_activa	VARCHAR(1)	S si esta activa la opcion del menu o n si no lo esta	No	No
		maam_estado	DATE	La fecha de registro de este acceso	No	No
ma_usuarios	Registra los usuarios que se crean para el manejo del sistema	maus_clave	VARCHAR(10)	Clave del usuario	Yes	No
		maus_usuario	VARCHAR(10)	Nombre corto del usuario	No	No
		maus_nombres	VARCHAR(40)	Nombre completo del usuario	No	No
		maus_modify_clave	VARCHAR(1)	Puede o no modificar la clave de usuario (s/n)	No	No
		maus_fecha_ingreso	DATE	Fecha de registro del usuario	No	No
		maus_fecha_expira	DATE	Fecha en que expira la clave del usuario	No	No
		maus_admin	VARCHAR(1)	Si es o no el administrador del sistema	No	No
		maus_bloqueado	VARCHAR(1)	Si esta o no bloqueado el usuario	No	No
		maus_numero_intentos	NUMERIC(2)	El numero de intentos que puede errar en el ingreso del sistema	No	No
		maus_ultimo_bloqueo	DATE	La fecha del ultimo bloqueo del usuario	No	No

		maus_estado	DATE	Ultima fecha de ingreso al sistema	No	No
		car_cod_maus	NUMERIC(2)	Código de la carrera	No	No
tbl_alumno	Registra la información personal del alumno	alumno_cod	NUMERIC(10)	Código de alumno	Yes	No
		alumno_apellido	VARCHAR(30)	Apellido del alumno	No	No
		alumno_nombre	VARCHAR(30)	Nombre del alumno	No	No
		alumno_lugar_nac	VARCHAR(20)	Lugar de nacimiento	No	No
		alumno_fecha_nacim	DATE	Fecha de nacimiento	No	No
		alumno_pais	VARCHAR(20)	País	No	No
		alumno_cedula	VARCHAR(10)	Cedula de identidad	No	No
		alumno_sexo	VARCHAR(1)	Tipo de sexo	No	No
		alumno_estado_civil	VARCHAR(1)	Estado civil	No	No
		alumno_direccion	VARCHAR(60)	Dirección domiciliaria	No	No
		alumno_telefono	VARCHAR(20)	Teléfono del domicilio	No	No
		alumno_celular	VARCHAR(20)	Teléfono personal	No	No
		alumno_colegio_procedencia	VARCHAR(50)	Procedencia institucional	No	No
		alumno_titulo	VARCHAR(25)	Titulo actual del alumno	No	No
		alumno_nombre_padre	VARCHAR(40)	Nombre del padre	No	No
		alumno_nombre_madre	VARCHAR(40)	Nombre del alumno	No	No
		alumno_direccion_1	VARCHAR(60)	Dirección	No	No
		alumno_telefono_1	VARCHAR(20)	Teléfono	No	No
		alumno_fecha_creacion	DATE	Fecha de registro del alumno	No	No
		alumno_email	VARCHAR(50)	Correo electrónico	No	No
		user_cod	NUMERIC(3)	Usuario que registro al alumno	No	No
		car_cod_al	NUMERIC(2)	Carrera a la que ingresa el alumno	No	Yes
		alumno_estado	VARCHAR(1)	Estado	No	No
		alumno_carne	VARCHAR(7)	Numero de carne	No	No
tbl_carrera	Registra la información de las carreras de la facultad de sistemas	carrera_codigo	NUMERIC(2)	Código de la carrera	No	No
		carrera_nombre	VARCHAR(80)	Nombre de la carrera	No	No
		carrera_direccion	VARCHAR(80)	Dirección	No	No
		carrera_telefono	VARCHAR(20)	Teléfono	No	No
		carrera_responsable	VARCHAR(50)	responsable de la carrera	No	No

		escuela_nombre	VARCHAR(100)	Nombre de la escuela a la que pertenece	No	No
		univ_nombre	VARCHAR(100)	Nombre de la institución	No	No
tbl_ciclo	Registra el ciclo(semestre) académico dentro del cual se trabajara en el sistema	ciclo_codigo	VARCHAR(5)	Código del ciclo	No	No
		ciclo_nombre	VARCHAR(30)	Nombre del ciclo	No	No
		ciclo_numero	NUMERIC(2)	Número de ciclo	No	No
		ciclo_tipo	VARCHAR(30)	Tipo de ciclo	No	No
		ciclo_fechainicio	DATE	Fecha de inicio del ciclo	No	No
		ciclo_fechafinal	DATE	Fecha de finalización	No	No
		ciclo_estado	VARCHAR(1)	Estado	No	No
		per_cod_ci	NUMERIC(4)	Periodo al que pertenece	No	Yes
tbl_hc_detalle	Registra las materias que toma el alumno en un determinado ciclo	hc_det_codigo	NUMERIC(20)	Código de la Hoja de Crédito Detalle	Yes	No
		profmatnp_cod	NUMERIC(12)	Código de la tabla Profesor, Materia, Nivel Paralelo, Pensum	No	Yes
		hc_matricula	NUMERIC(1)	Matricula que puede tener la materia (Primera, Segunda, Tercera)	No	No
		hc_estado	VARCHAR(1)	Control que verifica la aprobación del alumno de la materia	No	No
		hc_enc_cod_hc_det	NUMERIC(20)	Código de la Hoja de Crédito Encabezado	No	Yes
		hc_convalida	VARCHAR(1)	Control para la convalidación S/N	No	No
		hc_aux_nota	NUMERIC(8,3)	Sumatoria de todos los parciales de la materia correspondiente	No	No
		hc_aux_asistencia	NUMERIC(3)	Sumatoria de todos los registros de asistencia de la materia correspondiente	No	No
tbl_hc_enc	Registra la matricula del alumno en un determinado ciclo y pensum	hc_codigo	NUMERIC(20)	Código de la Hoja de Crédito Encabezado	Yes	No
		hc_fecha	DATE	Fecha de registro de la hoja de crédito	No	No
		hc_categoria	INTEGER	Categoria de Hoja de Créditos (Regular, Especial, Oyente)	No	No

		pen_cod_hc	NUMERIC(10)	Código del Pensum, con el que se va a asignar la Hoja de Créditos	No	Yes
		al_cod_hc	NUMERIC(10)	Código del Alumno	No	Yes
		user_cod	NUMERIC(3)	Control de usuario realizando el proceso	No	No
tbl_materia	Registra las asignaturas	materia_codigo	VARCHAR(10)	Código de la materia	Yes	No
		materia_nombre	VARCHAR(60)	Nombre de la materia	No	No
		car_cod_mat	NUMERIC(2)	Código de la carrera	No	No
		ni_cod_mat	NUMERIC(2)	Nivel de la materia	No	Yes
		materia_control	VARCHAR(1)	Control para asignación de prerrequisitos a Materias	No	No
		pen_cod_mat	NUMERIC(10)	Código del pensum	No	Yes
		mat_cod_dese	VARCHAR(10)	Código de usuario	No	No
tbl_nivel	Registra los niveles de estudio	nivel_codigo	NUMERIC(2)	Código del nivel	Yes	No
		nivel_nombre	VARCHAR(25)	Nombre del nivel	No	No
tbl_nivel_paralelo	Registra la asignacion de paralelos a los diferentes niveles de estudio	niv_par_cod	VARCHAR(3)	Clave Principal de la Tabla Relación Nivel Paralelo	Yes	No
		niv_cod_np	NUMERIC(2)	Clave Foránea de la Tabla Nivel	No	Yes
		par_cod_np	VARCHAR(2)	Clave Foránea de la Tabla Paralelo	No	Yes
tbl_notas_asistencia	Registra las calificaciones y la asistencia de los alumnos en cada una de las materias y cada uno de los parciales que existan en el pensum	na_codigo	NUMERIC(20)	Código del registro de calificaciones	Yes	No
		hedet_cod_na	NUMERIC(20)	Código de la hoja de detalle	No	Yes
		sist_par_cod_na	NUMERIC(4)	Código del sistema de evaluación a utilizar	No	Yes
		na_nota	NUMERIC(6,3)	Calificación	No	No
		na_asistencia	NUMERIC(3)	Faltas	No	No
		cod_al_aux	NUMERIC(10)	El código auxiliar del alumno	No	No
		protegido	VARCHAR(1)	protegido si o no	No	No
		cod_profmatnp_aux	NUMERIC(12)	Código auxiliar de la asignación de la materia	No	No
tbl_paralelo		paralelo_codigo	VARCHAR(2)	Código del paralelo	Yes	No

	Registra los paralelos que pueden asignarse a niveles	paralelo_nombre	VARCHAR(25)	Nombre del paralelo	No	No
tbl_parcial	Registra los parciales que pueden ser asignados a un sistema de evaluaciones	parcial_codigo	NUMERIC(2)	Código del Parcial	Yes	No
		parcial_desc	VARCHAR(40)	Descripción del Tipo de Parcial que se va a crear	No	No
tbl_pen_ciclo_sisteval	Registra los el sistema de evalucion que se dara a los alumnos en un determinado ciclo con un determinado pensum	pen_ciclo_sisteval_cod	NUMERIC(10)	Código de la asignación del pensum, ciclo y sistema de evaluaciones	Yes	No
		pen_cod	NUMERIC(10)	Código del pensum	No	Yes
		ciclo_cod	VARCHAR(5)	Código del ciclo	No	Yes
		sisteval_cod	NUMERIC(4)	Código del sistema de evaluaciones	No	Yes
		estado	VARCHAR(1)	Estado	No	No
tbl_pensum	Registra las asignaturas que se daran a los alumnos de acuerdo a la carrera	pensum_codigo	NUMERIC(10)	Código del Pensum	Yes	No
		pensum_desc	VARCHAR(60)	Descripción del Pensum	No	No
		pensum_estado	VARCHAR(1)	Estado en el que se encuentra el Pensum	No	No
		pensum_total_cre	NUMERIC(4)	Total de Créditos del Pensum	No	No
		car_cod_pensum	NUMERIC(2)	Código de la Carrera a la que pertenece el Pensum	No	No
		user_cod	NUMERIC(3)	Código del Usuario, que esta manipulando los datos, para control	No	No
tbl_periodo	Registra los periodos academicos anuales	periodo_codigo	NUMERIC(4)	Código del periodo	Yes	No
		periodo_nombre	VARCHAR(30)	Nombre del periodo	No	No
		periodo_estado	VARCHAR(1)	Estado	No	No
		car_cod_pe	NUMERIC(2)	Carrera a la que ingresa el alumno	No	No
tbl_prerequisito	Registra las materias que debe cumplir el alumno antes de tomar una materia superior	prer_codigo	NUMERIC(8)	Código del prerrequisito	Yes	No
		prer_mat_codigo	VARCHAR(10)	Código de la materia	No	No
		mat_cod_pre	VARCHAR(10)	Código de la materia que debe cumplir anteriormente	No	Yes

		pen_cod_pre	NUMERIC(10)	Código del pensum, para controlar el proceso de asignación de Pensum	No	No
tbl_prof_mat_np_pen	Registra la asignacion de profesores, niveles, paralelos, y materias en un determinado ciclo	cod_prof_mat_np_pen	NUMERIC(12)	Código de la asignación de niveles, paralelos, profesores, materias y ciclo pensum	Yes	No
		pro_cod	NUMERIC(10)	Profesor código	No	Yes
		mat_cod	VARCHAR(10)	Materia código	No	Yes
		np_cod	VARCHAR(3)	Nivel paralelo código	No	Yes
		num_creditos	NUMERIC(3)	Cantidad de créditos de cada materia	No	No
		pen_det_cod	NUMERIC(10)	Pensum detalle código	No	Yes
tbl_profesor	Registra los profesores que dictan clases en la carrera asignada	profesor_cod	NUMERIC(10)	Código del profesor	Yes	No
		profesor_cedula	VARCHAR(10)	Cedula de identidad	No	No
		profesor_nombre	VARCHAR(30)	Nombre	No	No
		profesor_apellido	VARCHAR(30)	Apellido	No	No
		profesor_sexo	VARCHAR(1)	Sexo	No	No
		profesor_fecha_nac	DATE	Fecha de nacimiento	No	No
		profesor_estado_civil	VARCHAR(1)	Estado civil	No	No
		profesor_email	VARCHAR(30)	Correo electrónico	No	No
		profesor_direccion	VARCHAR(40)	Dirección domiciliaria	No	No
		profesor_telefono	VARCHAR(20)	Teléfono	No	No
		profesor_celular	VARCHAR(20)	Teléfono personal	No	No
		user_cod	NUMERIC(3)	Código del Usuario que esta accediendo y modificando los datos	No	No
		car_cod_pro	NUMERIC(2)	Código de la carrera a la que da clases	No	Yes
tbl_sisteval	Registra la configuracion del sistema de evaluaciones a ser asignado a un ciclo	sisteval_codigo	NUMERIC(4)	Código del Sistema de Evaluación	Yes	No
		sisteval_desc	VARCHAR(60)	Descripción del Sistema de Evaluación con el que se va ha trabajar en el Pensum Encabezado	No	No
		sisteval_estado	VARCHAR(1)	Estado del Sistema de Evaluación, para controlar con cual sistema se esta trabajando en el ciclo	No	No

		sisteval_nota_min	NUMERIC(6,2)	Nota minima del sistema de Evaluación	No	No
		sisteva_nota_max	NUMERIC(6,2)	Nota máxima del Sistema de Evaluación	No	No
		car_cod_sisteval	NUMERIC(2)	Código de la Carrera a la pertenece el Sistema de Evaluación	No	No
		user_cod	NUMERIC(3)	Código del usuario	No	No
tbl_sisteval_parcial	Registra la asignacion de parciales a los sistemas de evalucion	se_par_codigo	NUMERIC(4)	Código de asignación de parcial al sistema de evaluaciones	Yes	No
		se_par_numero	NUMERIC(2)	Numero de los Parcial, Pacial 1, Parcial 2, que permite el orden de los datos	No	No
		se_par_min	NUMERIC(6,2)	Nota minima del Parcial	No	No
		se_par_max	NUMERIC(6,2)	Nota máxima del Parcial	No	No
		sisteval_cod_se_par	NUMERIC(4)	Código del Sistema de Evaluación, con el que se van a trabajar en los parciales	No	Yes
		parcial_cod_se_par	NUMERIC(2)	Código del Parcial, en el cual esta la descripción de los parciales	No	Yes

METODOLOGÍA

XP (eXtreme Programing) nace como nueva disciplina de desarrollo de software hace aproximadamente unos seis años, y ha causado un gran revuelo entre el colectivo de programadores del mundo. Con sus teorías ha conseguido el respaldo de gran parte de la industria del software y el rechazo de otra parte.

La programación extrema se basa en la simplicidad, la comunicación y el reciclado continuo de código, para algunos no es mas que aplicar una pura lógica.

Los objetivos de XP son muy simples: la satisfacción del cliente. Esta metodología trata de dar al cliente el software que él necesita y cuando lo necesita. Por tanto, debemos

responder muy rápido a las necesidades del cliente, incluso cuando los cambios sean al final de ciclo de la programación.

El segundo objetivo es potenciar al máximo el trabajo en grupo. Tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del software.

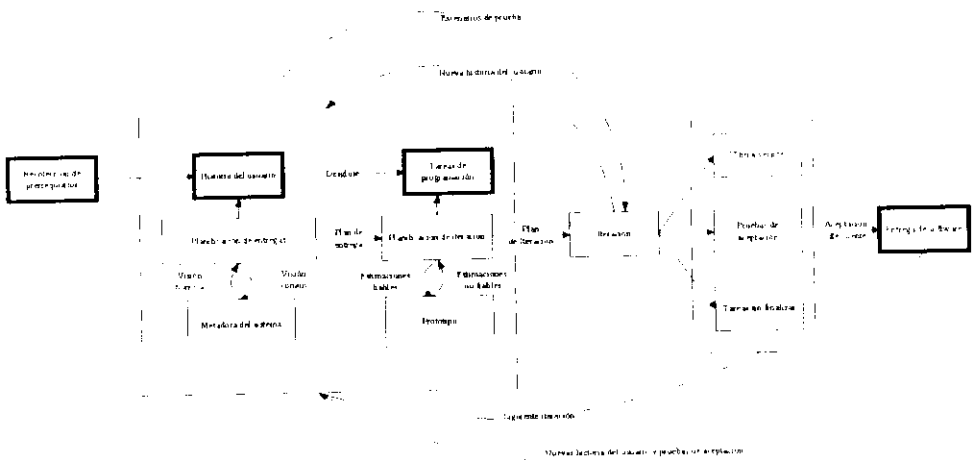
Existe la comunicación, simplicidad, retroalimentación,

- **Comunicación:** La programación extrema pone especial énfasis en el trabajo en equipo incluyendo gestores, desarrolladores y cliente dentro de un equipo multidisciplinar que persigue la producción de software de calidad.

- **Simplicidad**: mantener el diseño del sistema simple y claro.

- **Retroalimentación:** Dentro de la programación extrema la retroalimentación por parte del cliente se produce desde el primer día de trabajo, de este modo se estable como objetivo fundamental la satisfacción de las necesidades del cliente, entregándole el software que necesitar y cuando lo necesita.

Unas de la diferencias fundamentales de la programación extrema respecto a la otras metodologías es su respuesta segura al cambio en los requisitos por parte del cliente, incluso en fases avanzadas del ciclo de vida.



CAPITULO IV

4. VALIDACIÓN DEL PROYECTO

Ambato, 09 de enero de 2006

Ingeniero
Telmo Viteri
DIRECTOR DE LA ESCUELA DE SISTEMAS DE LA PUCESA
Presente.

De mi consideración:

La presente es portadora de un saludo cordial y a la vez informarle que se ha procedido a la validación del trabajo de disertación de la señorita: Catalina Elizabeth Morales Perrazo, titulado: "Desarrollo de una aplicación web para el registro y consulta de Notas en la escuela de Ingeniería de Sistemas de la PUCESA utilizando la Herramienta Visual Studio. NET " encontrando que el mencionado trabajo esta concluido a cabalidad, cumpliendo los objetivos trazados y funcionando plenamente en el cuarto Piso de la Institución.
Se destaca en el trabajo la investigación de la herramientas de programación como la aplicación de la misma en una solución puntual.

Atentamente,

Ing. Santiago Acurio M.
COORDINADOR ACADÉMICO
ESCUELA DE INGENIERIA DE SISTEMAS

Ambato enero 10, 2006.

Ingeniero
Telmo Viteri
DIRECTOR ESCUELA DE INGENIERIA DE SISTEMAS
Presente

De mi consideración:

Tengo a bien indicar que he revisado la parte práctica de la Disertación de Grado titulado "DESARROLLO DE UNA APLICACIÓN WEB PARA EL REGISTRO Y CONSULTA DE NOTAS EN LA ESCUELA DE INGENIERÍA DE SISTEMAS DE LA PUCESA UTILIZANDO LA HERRAMIENTA VISUAL STUDIO .NET", correspondiente a la señorita: Catalina Elizabeth Morales Perrazo, la cual se encuentra funcionando correctamente, cumpliendo con los objetivos planteados.

Particular que pongo en su conocimiento, para los fines pertinentes.

Atentamente,

Ab. Pablo Poveda Mora
SECRETARIO GENERAL PROCURADOR
Validador de la Disertación

4.1. COMPROBACIÓN DE LA HIPÓTESIS

Las implementación de un sistema del ingreso y consulta de notas para la Escuela de Ingeniería en Sistemas de la Pontificia Universidad Católica del Ecuador sede Ambato, permitirá simplificar y ahorrar tiempo en la verificación de los datos.

Variable independiente

Con la realización del Sistema de ingreso y consulta de notas

Variable Dependiente

Permitirá simplificar y ahorrar el tiempo en la verificación de los datos

A través del método lógico, se demuestra la hipótesis

A= Variable Independiente

B= Variable Dependiente

A ⟶ B

A: El sistema de información agilitara la consulta de notas para que los estudiantes se informen a tiempo.

B: Según el método Ponendo Pones: el software ayudará de mejor manera eficaz el ingreso y consulta de notas y ahorra el tiempo en la consulta de los datos.

CAPITULO V

5.1.CONCLUSIONES Y RECOMENDACIONES

5.1.1. CONCLUSIONES

Al culminar el desarrollo de la disertación de tesis se ha obtenido las siguientes conclusiones:

- El presente trabajo de investigación permitió diseñar una pagina web tanto el profesor como el estudiante tengan un acceso rápido y eficaz para poder consultar sus notas.

- El sitio web permite, al profesor ingresar sus notas y asistencia de las diferentes materias mientras que al alumno permite consultar sus notas sea por parcial, o un cuadro final de sus calificaciones.

- Con la implementación del sistema los alumnos podrán tener un servicio de consulta en el aspecto académico de una forma rápida y eficiente.

- La plataforma .Net permite tener una mayor facilidad en el desarrollo de aplicaciones web.

- La Pontificia Universidad Católica del Ecuador Sede Ambato no disponía de una pagina web sobre consulta de notas e ingreso de las mismas, el desarrollo de esta disertación ha permitido que dicho establecimiento tenga un mejor desenvolvimiento.

- Se debe utilizar computadores de alta capacidad con Sistemas Operativos actualizados (Win XP) para el "Ingreso y consulta de notas", ya que con el Sistema Operativo (Win 98) no se puede ejecutar porque no poseen el servicio de servidor web.

5.1.2. RECOMENDACIÓNES

- Aprovechar al máximo el proyecto después de haber implantado el sistema ya que presta un mejor servicio a estudiantes, profesores, y administrativos.

- Motivar a los estudiantes, continúen con el desarrollo de nuevos sistemas con tecnologías actuales e implementar en diferentes áreas académicas.

- Realizar nuevas aplicaciones aplicación en lenguaje basado en programación extrema, ya que esta programación permite retroalimentar cuando se tiene un problema como se avanza en el desarrollo del sistema.

- Estar pendiente de las nuevas tendencias tecnológicas y su implementación dentro del área educativa.

- Buscar tecnologías nuevas que trabajen bajo diferentes sistemas operativos.

- Mejorar la planificación cuando se tenga un proyecto dependencia el uno del otro para así poder evitar inconvenientes.

GLOSARIO DE TÉRMINOS

DBMS. (Data Base Management System): Es un conjunto de programas que se encargan de manejar la creación y todos los accesos a las bases de datos.

(SGBD) Sistema Gestor de Bases de Datos: es un conjunto de programas. procedimientos y lenguajes que proporcionan a los usuarios las herramientas necesarias para operar con una base de datos.

FNBC : forma normal de boyce

(A.B.D.): administrador de la base de datos son los encargados de diseñar la estructura de la base de datos y los responsables de que el sistema funcione correctamente

S.G.B.D: Es el Sistema gestor de la base de datos

BD: base de datos

ANY: significa que ese usuario tiene el privilegio para todos los esquemas en la Base de Datos.

ASP: Active Server Pages

XML: es un lenguaje de metamarcado que ofrece un formato para la descripción de datos estructurados.

CLR: Common Language Runtime

MSIL: Lenguaje intermedio de Microsoft

IDL: Lenguaje de definición de interfaces

IDE: es el entorno de desarrollo integrado

ASP: páginas Active Server

GUI: interfaz gráfico de usuario

POO: programación orienta a objetos

HTTP: Hypertext Transfer Protocol

SOAP: Simple Object Access Protocol para la comunicación entre programas

BIBLIOGRAFÍA

- **LIBROS:**

PETROUTSOS, Evangelos.
"La Biblia de Visual Basic .Net"

GONZÁLEZ SECO, José Antonio
"El Lenguaje de Programación C#"

SILER
"Visual Basic.Net" Edición Especial

GOMEZ JIMÉNEZ, Enrique .
Visual Basic.Net, Colección PASO A PASO

- **INTERNET**

http://www.microsoft.com/spanish/msdn/default
http://www.microsoft.com/spanish/msdn/articulos/default.asp
http://www.revista.unam.mx/vol.3/num1/art3
http://www.mindcraft.com/whitepapers/Sqlanywhere.html
www.mundovisioninc.com/wservices_es.php._web
www.lawebdelprogramdor.om/studio.net
www.microsoft.com/spanih/studio.net
http://www.microsoft.com/spanish/msdn/default
http://www.microsoft.com/spanish/msdn/visualbasic.net/default.asp
http://www.htmlpoint.com/sql/index.html
http://www.programacion.com/tutorial/csharp/2/
http://www.nmsdn.microsoft.com/net/
http://msdn.microsoft.com/net/ecma
http://www.wanadoo.es/

http://www.sscli.net

http://www.microsoft.com/net

http://www.desarrolloweb.com/articulos/1329.php?manual=48

http://perseo.dif.um.es/~fernan/clasifica.ppt

http://docentes.usaca.edu.co/wildiaz/BDII_02.html

http://www.algoritmia.net/articles.php?id=12

http://www.lafacu.com/apuntes/informatica/algorit_ordena/default.htm

http://decsai.ugr.es/docencia/ii/ii_mod_av_bd.html

http://www.itlp.edu.mx/publica/tutoriales/admonbasedat/tema2_1.htm

http://www.utem.cl/web/busqentabla.htm

http://www.desarrolloweb.com/manuales/9/

http://www.microsoft.com/spanish/msdn/articulos/archivo/151102/voices/fundamentos_xml.asp

http://www.gamarod.com.ar/articulos/xml.asp

http://www.desarrolloweb.com/articulos/1680.php?manual=54

http://msdn.microsoft.com/downloads/

http://pdf.rincondelvago.com/bases-de-datos.html

http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/dv_vjsharp/html/vjoriMicrosoftVisualJ.asp

http://msdn.microsoft.com/xml/default.asp

http://www.microsoft.com/net/developer/framework_com.asp

http://msdn.microsoft.com/msdnmag/issues/0900/Framework/print.asp

http://msdn.microsoft.com/msdnmag/issues/1000/Framework2/print.asp

http://msdn.microsoft.com/msdnmag/issues/0900/WebPlatform/WebPlatform.asp

http://www.microsoft.com/net/xmlservices.asp

http://msdn.microsoft.com/msdnmag/issues/0800/webservice/print.asp

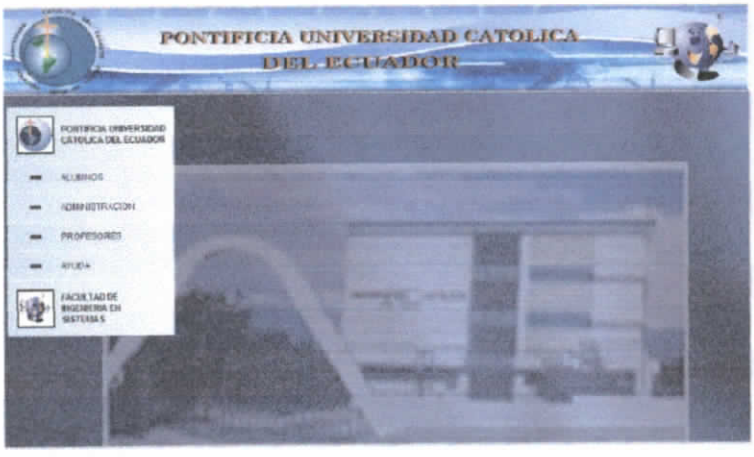
http://msdn.microsoft.com/msdnmag/issues/0300/soap/print.asp

MANUAL DE USUARIO

En el documento se detalla de cómo se debe acceder al sistema de ingreso y consulta de notas a continuación se detalla cada una de las pantallas del sistema propuesto:

MENÚ PRINCIPAL

Entre las cuales tenemos: alumnos, administrativo, profesores, ayuda..

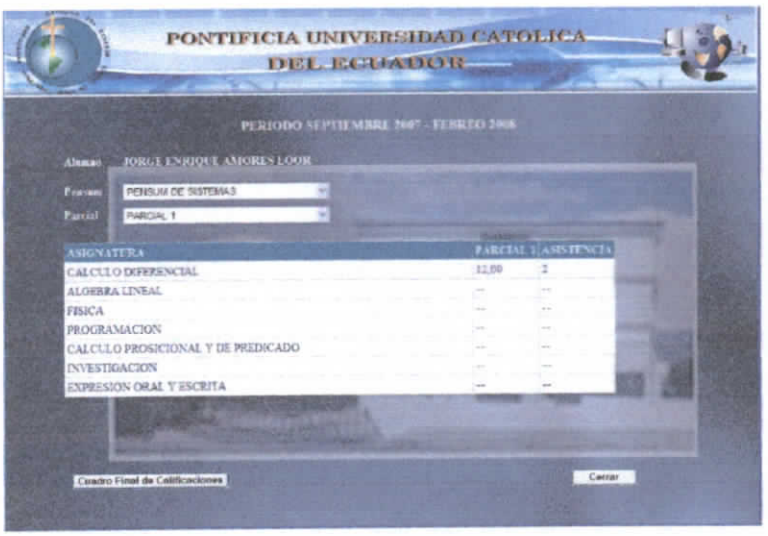


SECCIÓN ESTUDIANTES

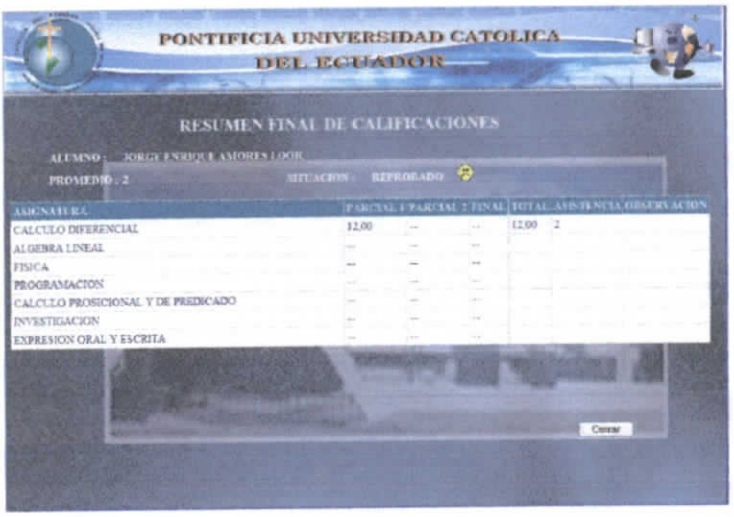
En esta opción nos permite ingresar el número de cédula del estudiante



En esta sección nos despliega las notas de cada uno de los estudiantes por parcial.

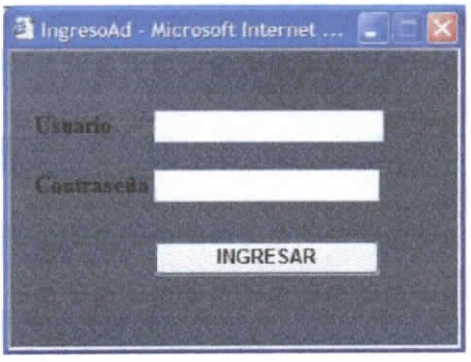


En esta pantalla despliega la consulta general de notas de los alumnos

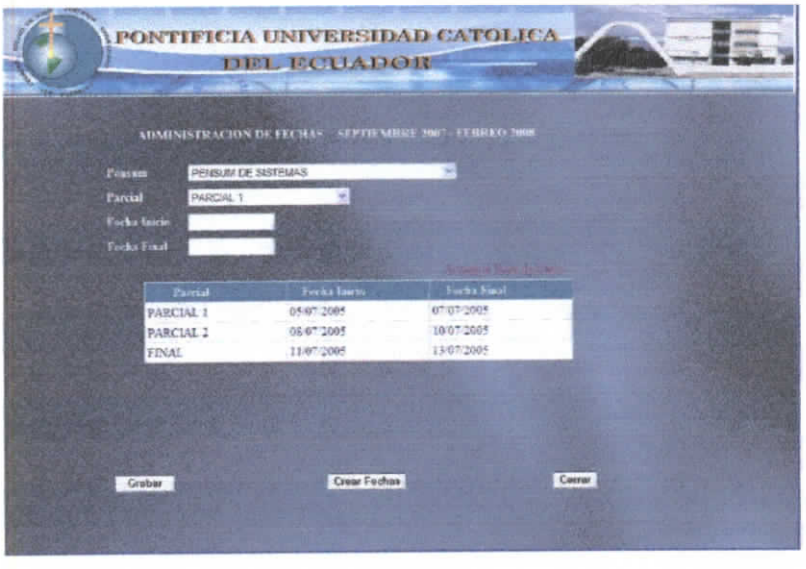


SECCIÓN ADMINISTRATIVO

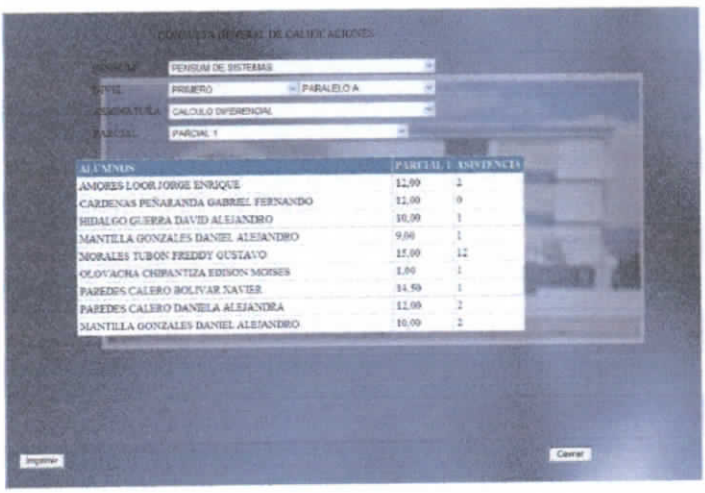
En esta sección permite ingresar el usuario y la clave



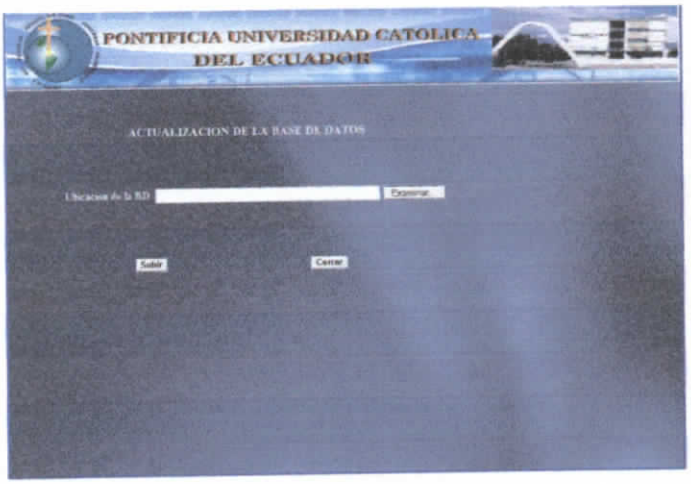
En esta opción al administrador da fechas para el ingreso de notas de cada parcial



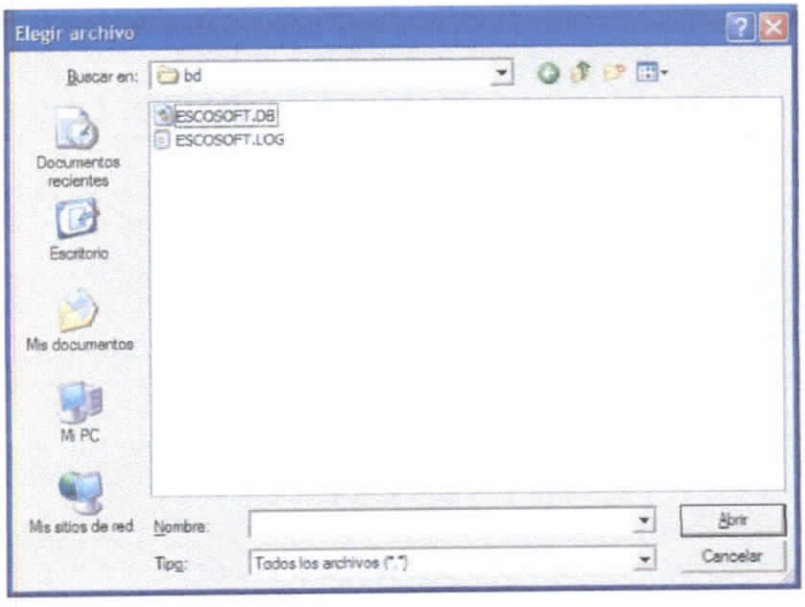
En esta pantalla el administrador puede consultar notas e imprimir las notas por parcial



En esta sección actualiza la base de datos mediante el botón examinar

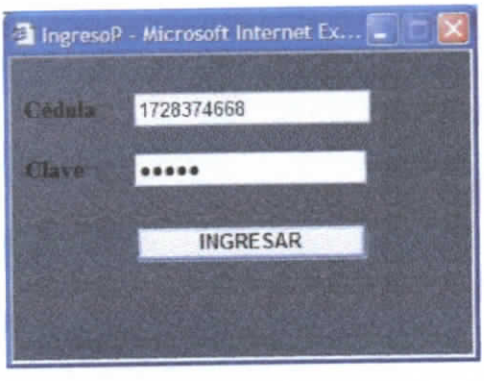


En esta pantalla permite subir la base de datos para actualizar

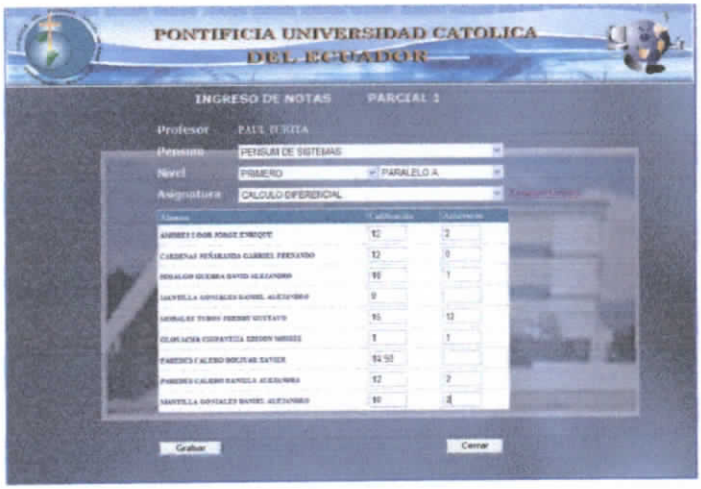


SECCIÓN PROFESORES

En esta sección el profesor puede ingresar el número de cédula y clave para poder ingresar notas, y la asistencia.



En esta sección se puede seleccionar la materia, el nivel, el parcial que va a ingresar y la asistencia.

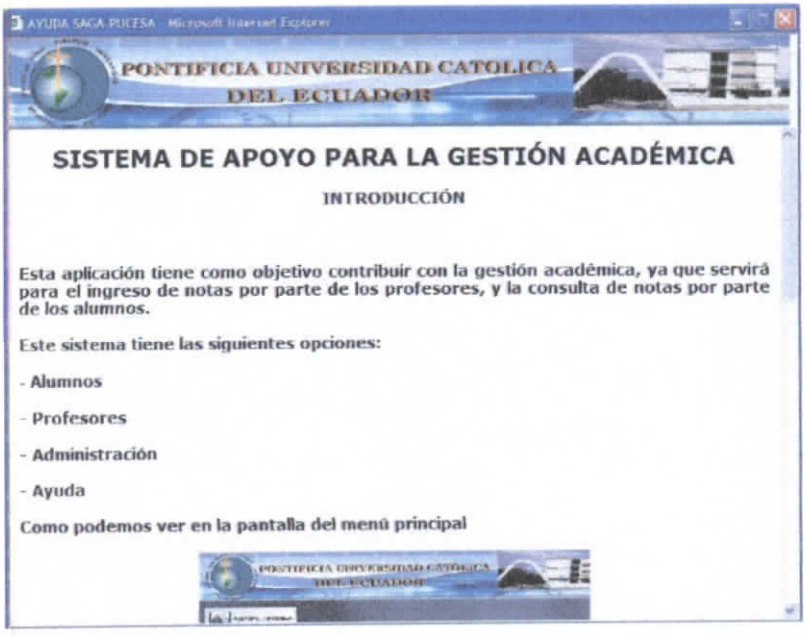


En esta sesión el profesor tiene una opción para poder consultar las notas ya ingresadas.



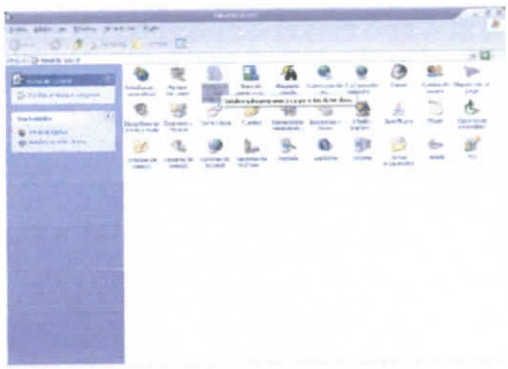
AYUDA

En esta sección nos da una explicación de cómo esta estructurado el sistema

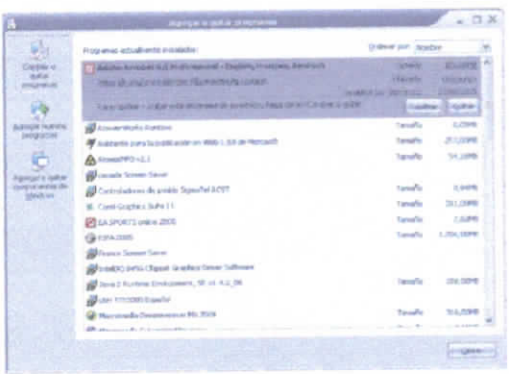


INSTALACIÓN DEL INTERNET INFORMATION SERVER.

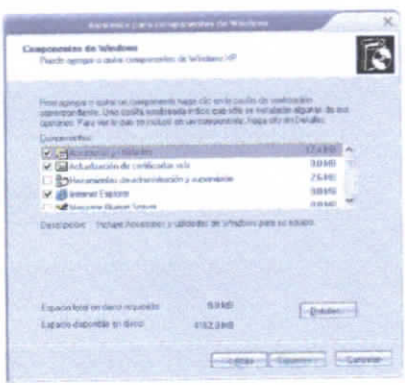
1. Inicio
2. Panel de control



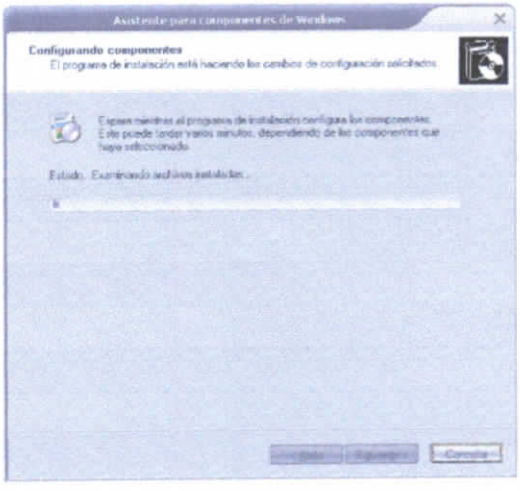
3. Agregar o quitar programas



4. Agregar o quitar componentes de Windows



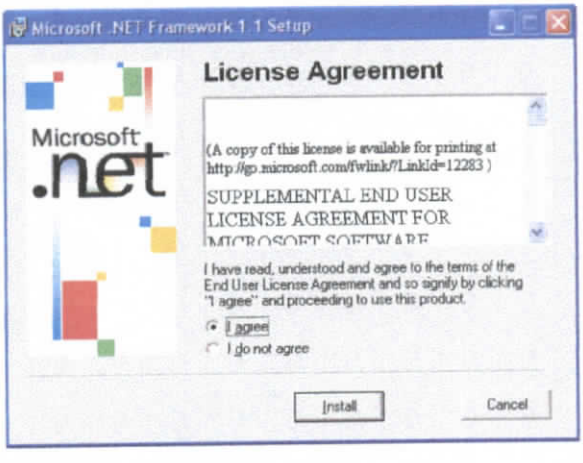
5. Activamos Servicios de Internet Information Server
6. Siguiente



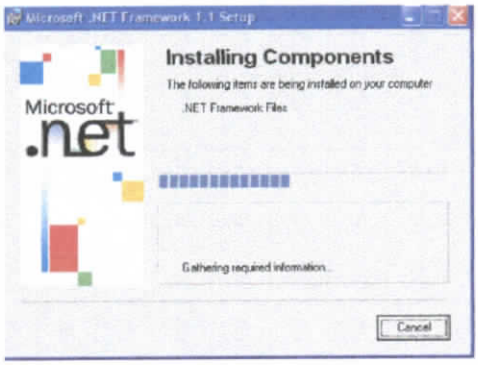
7. Y Finalizar

INSTALACIÓN DEL NET FRAMEWORK

1. Clic en el Setup



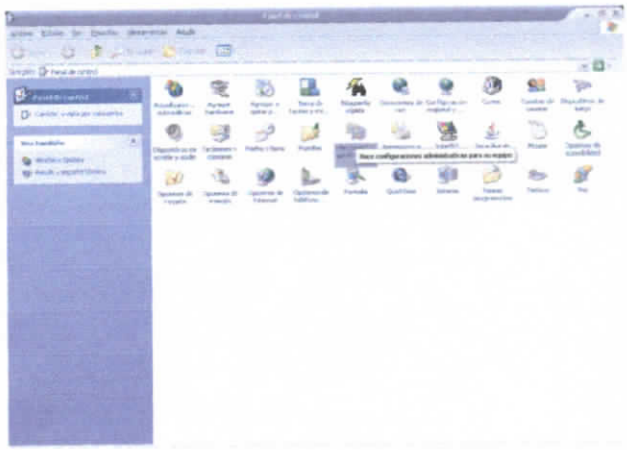
2. Luego Install



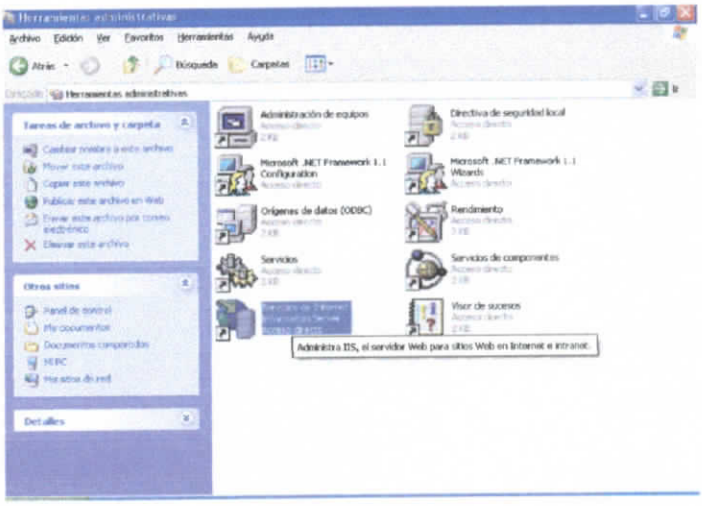
3. Finalizar

INSTALACIÓN DEL SISTEMA

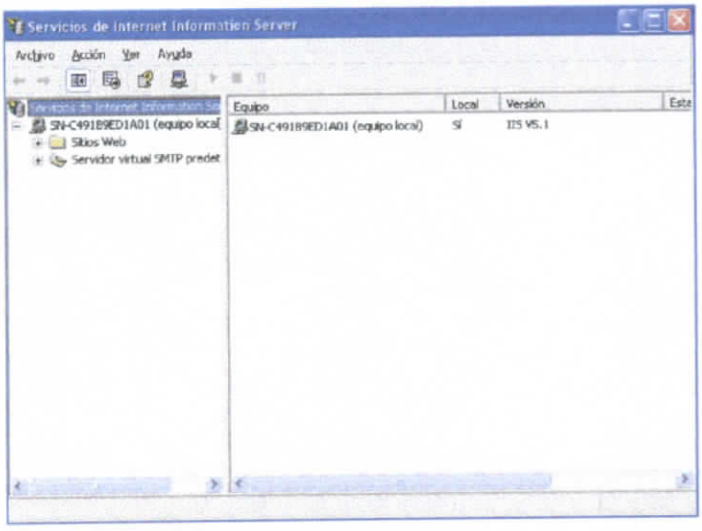
1. Copiar los archivos al disco duro
2. Luego para poder ejecutar el programa:
3. Inicio
4. Panel de control
5. Herramientas Administrativas



6. Servicios de Internet

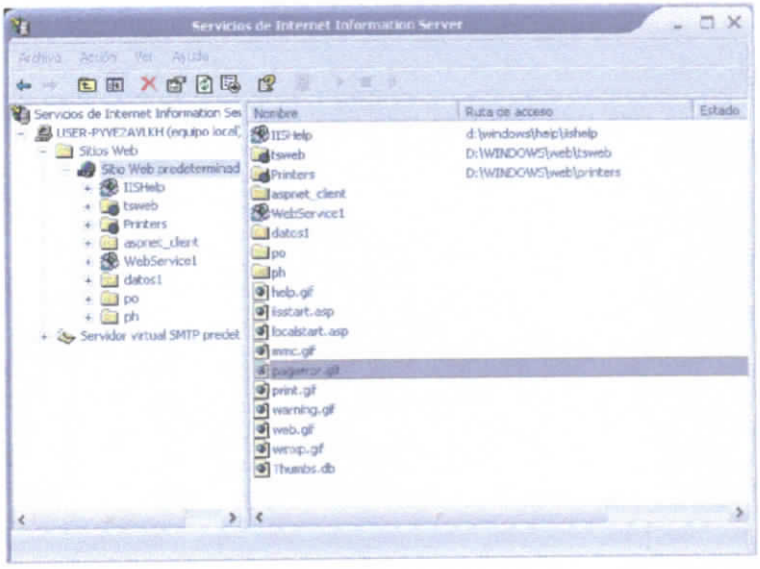


7. Servicios de Internet Information Server

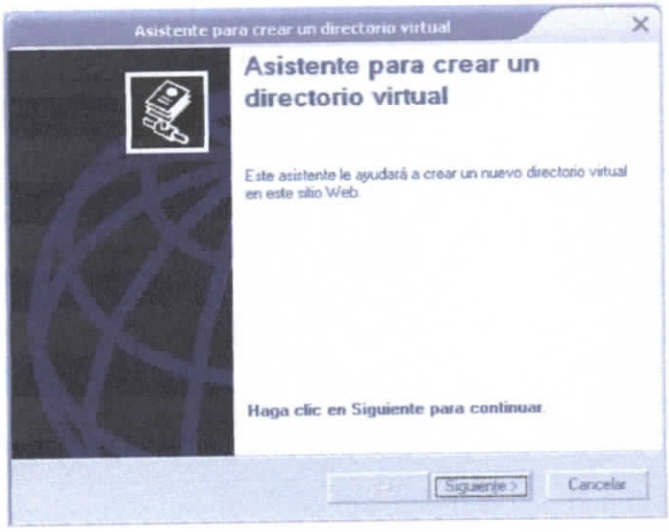


8. Sitios Web

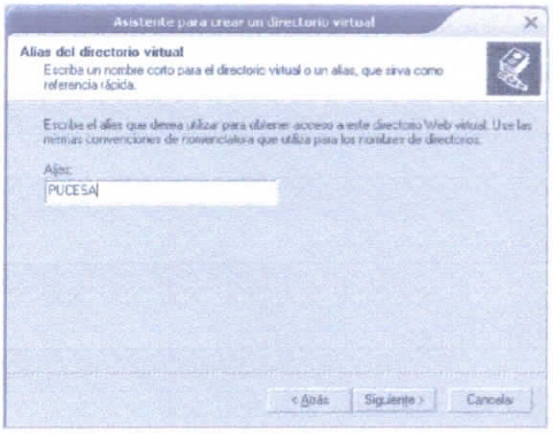
9. Sitios Web Predeterminado



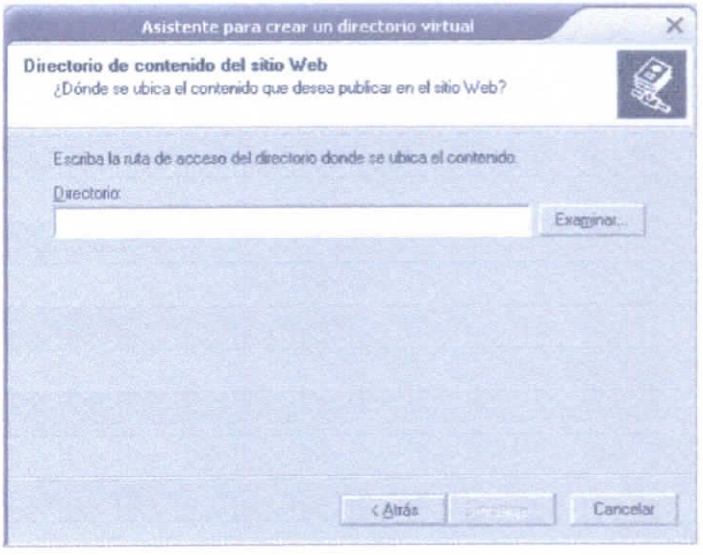
10. Crear una nuevo Directorio Virtual



11. Siguiente se da un alias llamado PUCESA

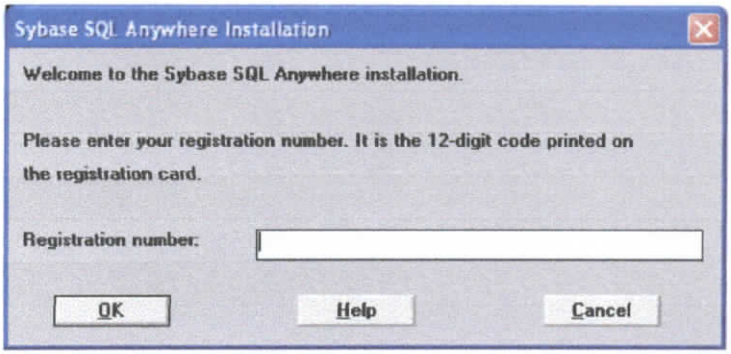


12. Siguiente

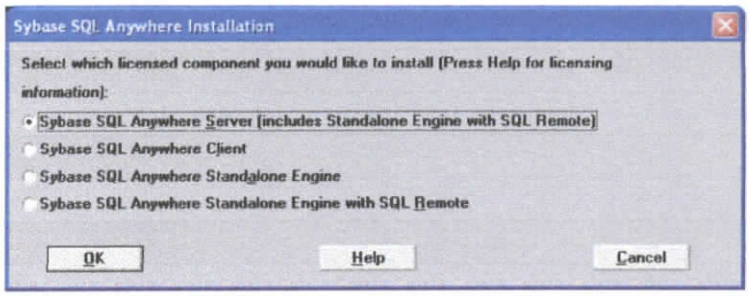


13. Clic en examinar
14. Direccionar a la carpeta copiada en el disco duro llamado Pucesa
15. Finalizar

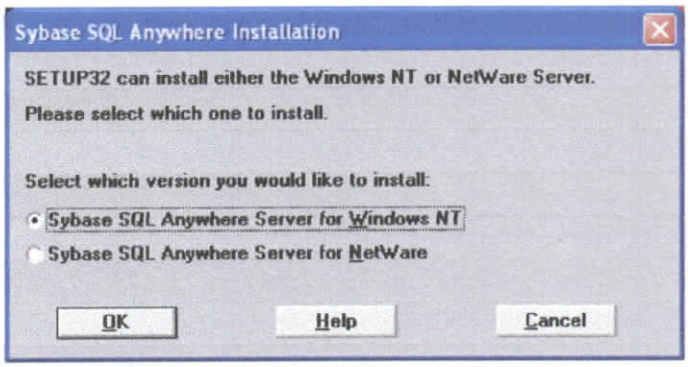
1. Clic en setup y ingresar el código



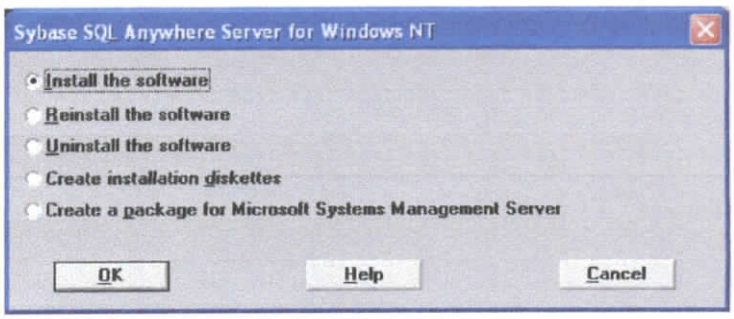
2. Clic ok



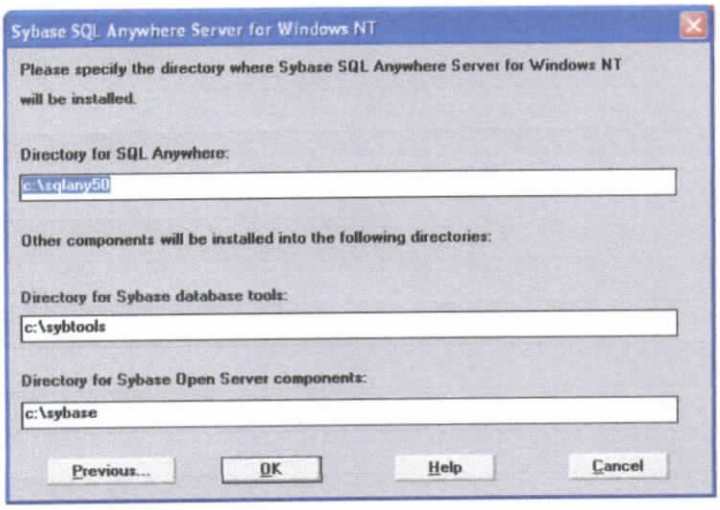
3. Clic en Ok



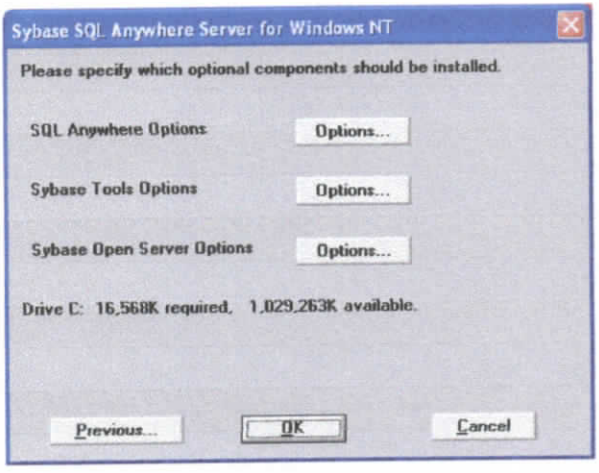
4. Selecciono instalación del software



5. Selecciono ha que unidad se va instalar



6. Clic en instalaciones de componentes



7. Clic en Ok para copiar todos los archivos del sistema



8. Por ultimo clic en ok y comienza la instalación

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR
PROGRAMAS ACADEMICOS AMBATO
BIBLIOTECA