

UNIVERSIDAD DE CIENCIAS COMERCIALES

FACULTAD DE INGENIERIA E INFORMATICA



**TESINA PARA OPTAR AL TITULO DE
INGENIERO EN COMPUTACION**

**Sistema de control de Registro de Notas y Matricula del
Colegio Anglicano en Puerto Cabezas**

AUTORES:

Alvaro Jose Bravo Ruiz

Jeysson David Dixon Foster

Erick Antonio Collado Saavedra

Alonso Antonio Kinsman Moody

TUTOR: MSc. Ing. Fausto Quiñonez

Managua, Nicaragua, C.A, Noviembre-2007

INDICE

	No. Pagina.
I. Dedicatoria.....	1
II. Agradecimiento.....	2
III. Introducción.....	3
IV. Objetivos.....	5
V. Justificación.....	6
VI. Antecedentes.....	7
VII. Marco Metodológico	
7.1. Análisis del sistema	
7.1.1. Requerimientos del negocio.....	8
7.2. Modelado del negocio	
7.2.1. Organigrama.....	10
7.2.1.1. Funciones del Personal.....	11
7.3. Planificación Estratégica de la Información (PEI)	13
7.4. Análisis del área del negocio.....	14
VIII. Proyectos Informáticos	
8.1. Análisis Costo Beneficios	
8.1.1. Viabilidad Técnica.....	15
8.1.2. Factibilidad Económica.....	17
8.1.3. Viabilidad Operativa.....	21
8.1.4. Viabilidad Legal.....	22
IX. Gestión De Proyecto.....	24
9.1. Personal.....	25
9.2. Problema.....	26

9.3.	Proceso.....	27
9.4.	Selección Del Modelo De Proceso (DRA).....	28
9.5.	Ámbito Del Software.....	30
9.6.	Métricas Del Software.....	31
9.7.	Estimaciones.....	32
9.8.	Análisis De Riesgo	
9.8.1.	Tipos De Riesgos.....	33
9.8.2.	Riesgos Específicos.....	35
9.9.	Gestión De La Configuración.....	36
X.	Requerimientos De La Plataforma Tecnológica (red).....	37
10.1.	Materiales y Componentes De La Red.....	39
10.2.	Características Red Área Local.....	40
10.3.	Topología Red Área Local.....	41
XI.	Fase De Desarrollo	
11.1.	Diseño Conceptual	
11.1.1	Casos De Uso.....	43
11.1.2	Diagrama De Clase.....	51
11.1.3	Diagrama De Componentes.....	54
11.1.4.	Diagrama De Secuencia.....	57
11.1.5.	Diagrama De Actividad.....	62
11.1.6.	Diagrama Entidad Relación.....	65
11.2	Diseño Lógico.....	66
11.3	Diseño Interfaz A Nivel Visual.....	66
11.4	Diseño Físico.....	67

XII.	Red De Área Local	
	12.1. Etapa De Realización.....	71
	12.2. Diagrama Red Área Local.....	72
XIII.	Conclusiones.....	73
XIV.	Recomendaciones.....	74
XV.	Anexos.....	75

DEDICATORIA

Dedicamos este logro al igual que muchos otros que hemos tenido en la vida como lo es este proyecto a Dios nuestro Padre, por brindarnos la sabiduría necesaria y la perseverancia para culminar con éxito nuestros objetivos propuestos.

A nuestros queridos padres de familia por el apoyo necesario que nos permitió culminar nuestros estudios y fortalecieron siempre con sus palabras y sabiduría.

Nuestros maestros por transmitirnos sus conocimientos día con día y al Colegio Episcopal Santísima Trinidad Anglicana quien fue objeto de nuestra inspiración para la realización del proyecto en miras del desarrollo de la Costa Atlántica de nuestro país.

AGRADECIMIENTOS

Agradecemos a todas las personas que nos brindaron en secreto su apoyo, para la realización de nuestro proyecto, especialmente al administrador del colegio Episcopal Anglicana Sr. Harold Gustavo Dixon, así mismo a la secretaria del mismo colegio Sra. Yessenia Susana Dixon Edwards por permitirnos realizarles las entrevistas pertinentes para la recopilación de la información necesaria del colegio.

Igualmente al Lic. Silvio Saballos por facilitarnos una breve reseña histórica de la Costa Atlántica (RAAN) y de la fundación de la iglesia anglicana y por consiguiente del Colegio Episcopal Anglicana en la región. Al Ing. Yader González por toda su colaboración, Msc. Irineo Moody y a la Ing. Belkys Iglesias por las horas disponibles y el apoyo valioso que nos brindaron. A todos ellos gracias por el tiempo desmedido entregado para lograr nuestras metas propuestas.

A nuestro tutor, Ing. Fausto Quiñones, por su grano de arena aportado en la realización del proyecto.

Sin obviar el apoyo moral y económico de nuestros padres para obtener el éxito profesional.

INTRODUCCION

La región autónoma del Atlántico Norte se caracteriza por ser una región multiétnica y pluricultural, ya que su sociedad está conformada por las etnias Miskitas, Mestizas, Criollos y Mayagnas siendo la etnia Miskita la de mayor predominio.

La Iglesia juega un papel determinante en la historia de la región, siendo la primera en fundarse la Iglesia Morava y posteriormente la Iglesia Anglicana. Estas Iglesias en si han jugado un papel determinante en la cultura regional, ya que con la llegada de los misioneros se instalan Iglesias y escuelas seculares pero con dirección cristiana.

La religión Anglicana funda su primera iglesia en 1954 y la primera escuela en el año de 1995 prestando sus servicios en este mismo año, llamándose "Colegio Episcopal Santísima Trinidad Anglicana". La escuela inicia sus labores con 8 aulas de clase y solamente en el primer grado. Incrementando su loable labor en los años subsiguientes, llegando a formar a más de 200 futuros profesionales en la región.

El colegio actualmente atiende en turnos regulares y en turno sabatino, a una población de unos 600 estudiantes que se instruyen en esta escuela con alrededor de 30 docentes. Además cuenta con todos los grados de nivel primario y secundario, haciendo notar que a partir del año 2003 se obtiene la primera promoción de bachilleres.

Como nota explicativa y a manera de justificación, se puede afirmar que los avances tecnológicos que se presentan a diario, ha facilitado la recopilación y

almacenamiento de información en todos los aspectos tanto gubernamentales, industriales, así como materiales de estudios.

El Colegio Anglicano, sujeto a estudio, a pesar de tener 12 años de fundación no cuenta con un sistema de cómputos que le permita llevar el control de las notas de los alumnos. Este aspecto incide en el de tiempo de búsqueda de información para alumnos, porque sus registros son llevados en archivos y a la hora de buscar cierta información se complica encontrar dichos datos por la cantidad de papeles en los que tiene que revisar.

OBJETIVOS

OBJETIVO GENERAL:

Implementar un modelo tecnológico de registro de matrículas y notas, en base a las necesidades reales del colegio Episcopal Santísima Trinidad Anglicana, mediante un sistema computarizado propio, que facilite las operaciones administrativas de dicho centro.

OBJETIVOS ESPECIFICOS:

1. Fortalecer el área administrativa, mediante un sistema computarizado diseñando una base de datos, para la recopilación de la información.
2. Facilitar eficientemente información requerida por padres de familia y alumnos.
3. Generar reportes de matriculas y notas cuando la dirección del colegio lo requiera.

JUSTIFICACION

En la costa Caribe Nicaragüense, el Sistema de Educación Publica cuenta con poca asistencia técnica en el campo de la aplicación informática, debido a la falta de recursos económicos y el apoyo de parte del gobierno.

La razón principal para desarrollar el presente proyecto es el proceso actual del Colegio Episcopal Santísima Trinidad en cuanto a la realización de las matriculas y registros de notas. Todo se procesa manualmente, registrándose en un libro diario el cual provoca pérdida de tiempo para los padres de familia que llegan a matricular sus hijos, y por otro lado la información no se guarda con seguridad ya que los archiveros donde se guarda la información no cuenta con ningún medio de protección.

Como todo proceso de servicio al cliente, puede apoyarse con un sistema de registro que permita captura, búsqueda y reporte de información oportuna, segura y sobre todo eficiente, para lograr la calidad necesaria y hacer que el cliente quede satisfecho por el servicio mismo. El proyecto vendrá a reducir el tiempo que se toma en hacer una matrícula, una revisión y/o control de calificaciones y mejorará la manera de manejar la información.

ANTECEDENTES

El Colegio Episcopal Santísima Trinidad de la Región Autónoma del Atlántico norte, que no cuenta con un sistema apropiado para el registro de matrículas y de notas de los alumnos.

Desde la fundación del Colegio y aún investigando la actividad de centros de estudio similares en Puerto Cabezas, se observa que nunca se ha realizado ningún tipo de estudio que analice la situación de un colegio y busque recursos tecnológicos para mejorar sus procesos administrativos y financieros. Las razones son muchas, por tanto no existe ningún tipo de antecedente en el cual podamos tomar como base para el seguimiento de un estudio como este, que es el primero en su categoría.

Es esta la razón por el cual, a partir de este estudio se desarrolló todo un sistema informático, que permita al personal administrativo y de dirección agilizar procesos como:

- (1) servicios de matricula
- (2) informes de calificaciones
- (3) control y registro de notas

Que permitan una atención ágil y eficiente a padres de familia y estudiantes que solicitan información requerida, manteniendo una base de datos almacenado en formato digital donde se hará una recopilación de todo el historial del estudiantado.

Uno de los beneficios más importantes que permitirá la implementación de este sistema de administración de notas y matricula, además de llevar un control eficaz, es que le permitirá estar a la altura al auxiliarse de tecnología de información actualizada, siendo más competitivos, más apreciados, más elegibles por los padres de familia para la formación de sus hijos.

MARCO METODOLOGICO

I. Análisis del sistema.

1. Requerimientos del negocio:

De acuerdo a la necesidad que presenta el Colegio Anglicano de no contar con un sistema automatizado para el registro de notas y matriculas de los alumnos que les permita administrar con eficiencia sus datos, atrasa actividad al realizarlas manualmente y guardarlos en archiveros de oficina por la secretaria del centro de estudios, después que los maestros entregan las actas de notas y tienen el visto bueno de administración.

Por lo tanto el presente Sistemas de Registro, permitirá controlar el registro de las matrículas que se realizan anualmente así mismo el registro de las notas directamente en la PC, con la intervención de la secretaria, guardando toda la información en una Base de Datos en el servidor. Al contar con un sistema automatizado el Colegio Anglicano podrá realizar los procesos operativos en menor tiempo.

Como habrán observado, este sistema está diseñado exclusivamente para que el encargado del departamento de Secretaría, opere de modo que sea más eficiente y eficaz.

Para la implementación de este sistema se hace necesario de ciertos requisitos de Hardware y Software, tanto para la PC cliente como la PC Server, en un sistema cliente-servidor.

La PC que funcionará como servidor tendrá los siguientes requisitos mínimos:

- Procesador INTEL CELERON de 2.8GHZ o mayor.
- Memoria de 512MB.
- Disco duro de 80GB.

Las PC's que funcionarán como clientes tendrán los siguientes requerimientos mínimos:

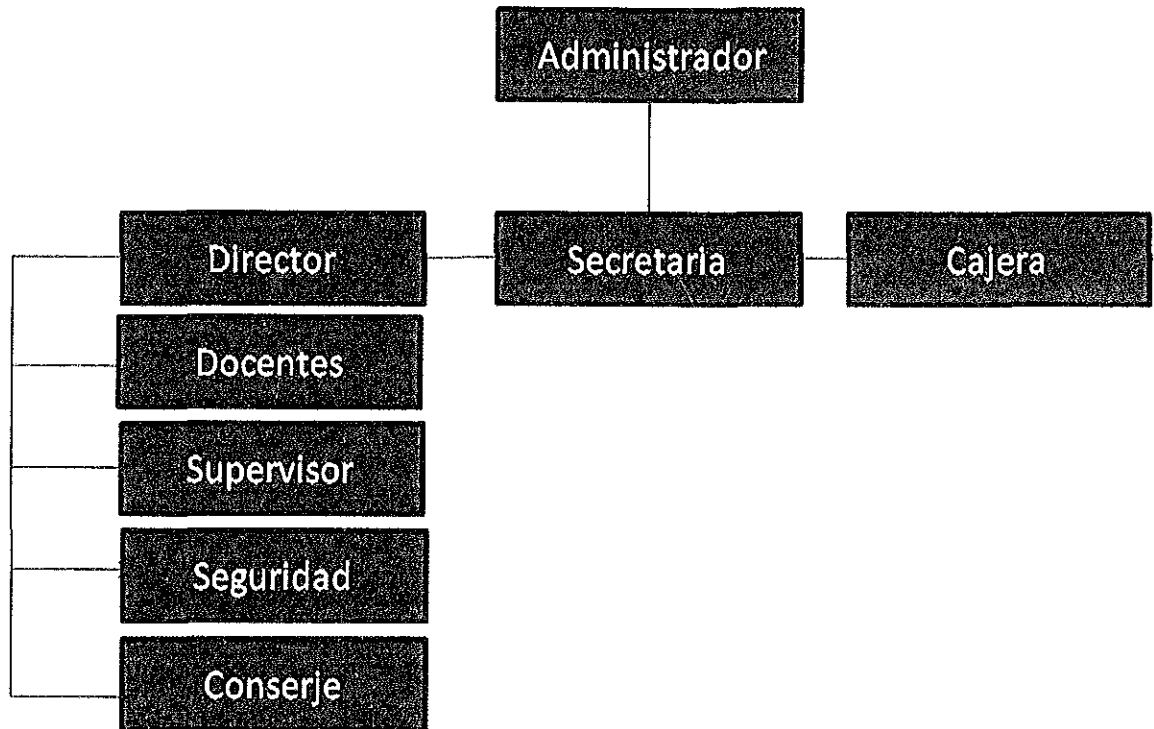
- Procesador INTEL CELERON de 2.8GHZ.
- Memoria de 512MB.
- Disco duro de 80GB.

Los materiales y componentes para la instalación de la red de área local son los siguientes:

- Switch 8 puertos.
- Cable Utp categoría 5.
- Conectores Rj 45.

MODELADO DEL NEGOCIO

Organigrama del Colegio Anglicano



Funciones del Personal

Administrador

Es el responsable del Colegio y encargado de llevar el control de registro de asistencia del personal de dicho centro velando el cumplimiento del horario establecido del Trabajo.

Secretaria

Es la encargada de recibir e informar cordialmente a los padres de familia, tener bajo su custodia y responsabilidad los documentos del centro, como libros de registro de notas y matrículas.

Director

Es el responsable del buen funcionamiento del Colegio le corresponde la elaboración, seguimiento y control del plan de trabajo del centro, responsable de labor ante las delegaciones departamental.

Cajera

Encargada de recibir el pago de aranceles o cualquier otro tramite que tenga que ver con el giro del negocio.

Docentes

Responsable por el buen funcionamiento del grupo a su cargo, garantizar la participación de los alumnos en el aula de clase así como velar por la salud y educación.

Supervisor

Garantizar que los alumnos se encuentren en su aula de clase cuando la maestra este impartiendo su clase.

Seguridad

Encargado de la seguridad del Colegio.

Conserje

Encargada del aseo del Colegio.

P.E.I

La PEI, es una poderosa herramienta de diagnóstico, análisis, reflexión y toma de decisiones colectivas, en torno al quehacer actual y al camino que debe recorrer en el futuro las organizaciones e instituciones, para adecuarse a los cambios y a las demandas y lograr el máximo de eficiencia y calidad de sus prestaciones.

La planificación del proyecto se realiza para la obtención de un bosquejo de lo que el sistema va a realizar, los procesos y resultados que se esperan.

Los objetivos generales del PEI:

Una estrategia que incorpore la calidad debe de tener en cuenta los siguientes elementos e incluir objetivos relacionados con ellos.

Definir los Objetivos y Metas del Negocio.

1. Enfoque hacia el cliente.
2. Énfasis en la prevención y en la mejora de los procesos.
3. Analizar la automatización correcta de las metas.

2. Análisis del área de negocio

Al requerir alguna información respecto a las matrículas escolares o notas obtenidas por los alumnos ya sea para información general o para la generación de reportes por alguna parte interesada dentro o fuera del centro, se tiene que hacer manualmente lo cual provoca una pérdida de tiempo innecesaria que ocurre en la falta de eficiencia y la eficacia de los recursos humanos en otras áreas del centro.

Cuando se realizan las matrículas al iniciar el año escolar los padres de familia llegan al centro con la documentación necesaria del hijo a matricular y se presenta ante la secretaria quien es la única responsable para el registro de la matrícula de los alumnos y el archivo de la misma.

Se llenan los datos correspondientes y se le asigna el grupo al que pertenecerá el alumno, docente guía, el número de clases a llevar, el aula de clases y el turno. Toda esta información se registra manualmente y posteriormente es guardada por la misma secretaria en los archivos del colegio.

Para el registro de las notas, se realizan las debidas evaluaciones mensuales de parte de los maestros hacia los alumnos, registrando cada nota obtenida por estos en la hoja de evaluaciones y al terminar el semestre se pasa la nota final del semestre a la secretaria que lo registra en un libro de actas y es guardado en archivos.

Proyectos Informáticos

1. Análisis Costo Beneficios:

Viabilidad Técnica

Plantea que recursos tanto físicos como recursos humanos se necesitaron para el desarrollo del sistema de información, lo existente actualmente en el colegio y las recomendaciones de hardware y software para el montaje del sistema en dicho centro de enseñanza.

Recurso Existente	Recurso	Características
3 PC	Hardware Y Software	Procesador 460 MHz Memoria RAM 64 MB Disco Duro 10 GB Floppy Drive Lector de CD Teclado Mouse Monitor de 15" Sistema Operativo Windows 2000 Profesional
1	Impresora	Impresora a Color Hp 3042 Deskjet

Tabla No.1

Requerimientos de Hardware para el óptimo funcionamiento del sistema

Descripción	Cantidad	Características
PC Cliente	2	Procesador 1.6 GHz Memoria RAM 512 MB Disco Duro 80 GB Floppy Drive Lector de CD Teclado Mouse Monitor SVGA CRT 15"
Servidor	1	Procesador 2.8 GHz Memoria RAM 512 MB Disco Duro 160 GB Floppy Drive Lector y quemador de CD Teclado Mouse Monitor SVGA CRT 15"
Impresora	1	Impresora Laser Jet Hp 1018
UPS	3	Respaldo de 5-20Min 5 Tomas Estabilizador Integrado

Tabla No.2

Requerimientos de Software para el óptimo funcionamiento del sistema

Descripción	Cantidad	Características
Microsoft Windows XP Profesional	2	Versión en Español ServiPack 2
SQL Server 2005	3	
Visual Basic.NET 2005	1	
Norton Antivirus	3	2007

Tabla No.3

Factibilidad Económica

Indica los costos de equipos de computo a adquirir, las licencias de los software a utilizar y Recursos Humanos.

Equipos	Cantidad	Costo Unitario \$	Costo Total \$
Impresora Laser Hp 1018	1	94.00	94.00
PC Servidor	1	369.00	369.00
PC	2	354.00	708.00
UPS	3	38.00	114.00
TOTAL	7	855.00	1,285.00

Tabla No.4

Ver anexo (cotización No. JL2911007)

Software:

Sistema operativo de la red	Cantidad	Costo Unitario \$	Costo Total \$
Windows XP	3	164.00	492.00
Norton Antivirus	3	49.00	147.00
TOTAL	8	213.00	639.00

Tabla No.5

Ver Anexo (Proforma No. 00-0000043222)

Presupuesto para red de área local:

Materiales	Cantidad	Costo Unitario \$	Costo Total \$
Switch	1	60.00	60.00
Cable UTP	75 (mts)	0.26	19.5
Conector RJ45	10	0.25	2.50
Canaletas ¾	75 (mts)	4.75	356.25
Codos	3	1.17	3.51
TOTAL	16	66.43	441.76

Tabla No.6

Ver Anexo (Cotización No. C00666, cotización No. JL2911007)

Recursos Humanos

Personal	Cantidad	Costo Unitario \$	Costo Total \$
Costo del Software	16 Formularios	75.00	1,200.00
Secretaria	1	134.40	134.40
Diseñadores de red	1	45	45.00
Estudio de preinversión	4	25	100
TOTAL	10	364.40	1479.40

Tabla No.7

Materiales de oficina

Descripción	Cantidad	Costo Unitario \$	Costo Total \$
Guillotina Merletto	1	17.33	17.33
Papel bond	1 caja	38.86	38.86
TOTAL	2	56.19	56.19

Tabla No.8

Ver Anexo (Factura Proforma No. 0000029958)

Herramientas de trabajo

Descripción	Cantidad	Costo Unitario \$	Costo Total \$
Ponchadora RJ 45	1	20.00	20.00
Tester de red	1	65	65
TOTAL	2	85.00	85.00

Tabla No.9

Ver Anexo

PLAN GLOBAL DE INVERSIONES

RUBROS	MONTO GLOBAL \$
A. INVERSIONES FIJAS	
A.1. EQUIPOS	1,285.00
A.2 HERRAMIENTAS DE TRABAJO	85.00
A.3 MOBILIARIO Y EQUIPO DE OFICINA	56.19
A.4 SOFTWARE (DESARROLLO)	1,200.00
B. INVERSIONES DIFERIDAS	
B.1. ESTUDIOS DE PREINVERSION	100.00
B.2 LICENCIAS	639.00
B.3 RECURSOS HUMANOS	179.40
B.4. DESARROLLO RED ÁREA LOCAL	441.76
INVERSIONES TOTALES	3,986.35

VIABILIDAD OPERATIVA

Comprende la capacitación de los Recursos Humanos, que se encargara del manejo de los procesos con el software diseñado.

Dicha capacitación tendrá una duración de 30 días, con un periodo de 2 horas por día semanalmente excluyendo los fines de semana, impartida por los desarrolladores del software, el costo de esta capacitación ya viene incluido en el costo total del proyecto.

El personal a capacitar será la secretaria del Colegio Anglicano quien ya cuenta con conocimientos básicos en el área informática el cual permitirá una mayor asimilación del manejo del sistema y disminuyendo así el riesgo de un mal manejo del mismo.

VIABILIDAD LEGAL

En la viabilidad legal se contemplan la existencia de los derechos y restricciones legales, en la realización de un sistema informático, derechos de autor, licencias para el software y otros. Para el desarrollo de algún software es necesario la elaboración de un contrato donde se reflejan las cláusulas como garantía de cumplimiento que establecen el respaldo para el cliente, económicamente y a través de aseguradoras para garantizar la finalización del proyecto a como se había contemplado. **Ver Anexo (No. 4)**

El cliente y los desarrolladores del software, irán ajustando el software a las necesidades de la empresa por lo cual esta se compromete al pago de honorarios correspondientes.

En caso de las garantías de cumplimiento por aseguradoras, el cliente puede establecer una cláusula de penalización que deduce al desarrollador del sistema por atrasos en la entrega del producto. Son del 1% sobre el monto del contrato por el periodo de tiempo vencido.

También se debe contemplar una cláusula de adendum cuando el cliente reclame funcionalidades para el sistema que no hayan sido contempladas en términos de referencia, los desarrolladores se reservan el cobro de un monto según cambios aplicados al software. Al igual la exoneración a los desarrolladores del software del mal manejo de base de datos, servidores, equipos de comunicación respaldo y dispositivos que son responsabilidad del departamento de sistemas del cliente.

Establecer garantías del producto, ya sea en común acuerdo con el cliente o desarrolladores del software, comprometiéndose a la realización de correcciones sin costos adicionales para el cliente.

GESTION DE PROYECTO

Para una gestión de proyecto eficaz implica la planificación, supervisión y control del personal, de los procesos y cualquier evento que pueda ocurrir mientras se desarrolla el software. Si el personal involucrado no cumple con su trabajo específico, no se logrará el éxito en la gestión de proyecto.

Por tanto los pasos para la gestión del desarrollo de un software son los siguientes:

- **Personal:** organizarse para el desarrollo del trabajo de sistema con efectividad.
- **Producto:** la determinación del alcance del producto mediante la comunicación con el cliente.
- **Proyecto:** planificar tomando en cuenta la estimación del esfuerzo y el tiempo para el cumplimiento de las tareas.

El gestor de proyecto debe de hacer lo correcto al momento de la estimulación del personal para la colaboración efectiva, poniendo su mayor empeño en las necesidades del cliente y en la calidad del producto.

PERSONAL

El personal está calificado para el desarrollo del software, utilizaremos por tanto el modelo de madurez que desarrolla la selección, gestión de rendimiento, entrenamiento, retribución, el desarrollo del diseño de trabajo y organización, cultural y el espíritu de equipo.

- **Los gestores:** que poseen la capacidad de desarrollo y la técnica para el diseño de la aplicación del software son:
 - Alvaro Bravo Ruiz.
 - Erick Collado Saavedra.
 - Jayson Dixon Foster.
 - Alonso Kinsman Moody.
- **Clientes:** son los que brindan la información necesaria de sus necesidades a los desarrolladores, para el diseño del software.
- **Usuario final:** es el encargado de controlar los diferentes procesos del software que será "la secretaria del centro escolar".

PROBLEMA

Nuestro problema es la necesidad del diseño y desarrollo de un software que realice el control del registro de las matriculas y las notas del alumno y la generación de reportes de las mismas del colegio que será manipulado por la secretaria del colegio.

Los procesos que realizara la secretaria en el sistema será:

1. La digitalización de información
 - 1.1 Realización de matrícula del alumno.
 - 1.2 Ingreso de notas al sistema.
2. Operaciones de la generación de reportes impresos
 - 2.1 Reporte de matrícula de los alumnos.
 - 2.2 Reporte de notas del alumno.

La generación de estos reportes será al momento de requerirlo por el mismo alumno y la dirección.

PROCESO

El gestor se encarga de la elección del proceso adecuado; quien realiza y concluye con la fase de las entrevistas que permite la recopilación de los datos necesarios y de los procesos que se realiza actualmente en el colegio.

Implementación de software prototipo de acuerdo a los datos obtenidos de las necesidades del colegio con una interfaz eficiente y dinámica, desarrollando un sistema ajustado a la necesidad del Colegio Anglicano.

Para el cual se utilizará lenguajes de programación como son:

Windows server 2005

SQL Server 2005

Utilizando la metodología orientada a objetos de acuerdo a las necesidades que presenta el problema del centro, que en su necesidad final requiere de Base de Datos relacionales.

SELECCION DEL MODELO DE PROCESO

Se seleccionará el método DRA (Desarrollo Rápido de Aplicación) siendo esta una aplicación de alta velocidad en el modelo final, secuencial ideal para construcciones basado en componentes.

El modelo DRA comprende de 5 fases:

- Modelo de gestión.
- Modelo de datos.
- Modelo de proceso.
- Generación de aplicaciones.
- Prueba y entrega.

Modelo de gestión:

Son las necesidades del Colegio Anglicano para el proceso del registro de las matriculas, registro y control de las notas de los alumnos de manera practica con eficiencia.

Modelo de datos:

Es donde se generan los datos del alumno realizados en la matricula y los datos de las notas obtenidas por los alumnos del colegio mediante reportes de:

-matricula

-notas

Modelo de procesos:

La generación de todos los datos será realizado por la secretaria.

Generación de aplicaciones:

Todos los datos que ingresan al sistema serán guardados en la Base de Datos creada en SQL.

Prueba y entrega:

Todos los procesos que se realizan en el sistema serán por la secretaria.

AMBITO DEL SOFTWARE

Nuestro sistema por sus características puede ser utilizado por otro centro de estudios que cuente con las mismas características organizacionales y tengan las mismas necesidades, no es exclusivamente para el Colegio Anglicano, contando con que en la RAAN (**Región Autónoma Atlántico Norte**) los colegios mantienen el mismo perfil.

El sistema será para el departamento de secretaria, el acceso al sistema solamente será autorizado para la secretaria y el director. Con este software se tendrá acceso rápido y eficaz a la información de los datos guardados en ella, en un corto plazo de tiempo e información segura. El almacenamiento, portabilidad de la información será segura, así mismo dará seguridad en la entrada de los datos.

Contando con una interfaz amigable y dinámica, teniendo objetivos que cumplir como:

- La seguridad de usuario.
- Seguridad de datos y la administración.
- Rapidez y elaboración de reportes
- Administración de datos personales.

METRICAS DEL SOFTWARE

La medición del software, permite la mejora en el seguimiento, control, evaluación y planificación del software a los gestores y/o desarrolladores.

Cuando hablamos de la medición de un software implica alejarnos de lo que comúnmente conocemos de la ingeniería de software, pero que posee razones concretas del porque realizarlas.

Entre ellas podemos mencionar, la indicación de la calidad del producto, calidad de los desarrolladores del producto y para la evaluación de los beneficios (productividad y calidad), uso de métodos nuevos y herramientas de ingeniería de software.

Así mismo ayuda en la justificación del uso de nuevas herramientas de formación adicional.

ESTIMACIONES

Para el cálculo del costo del sistema se hará mediante un recuento de la cantidad de formularios codificados que contiene la aplicación, el total de formularios en el sistema es 16, los cuales contienen la aplicación y reportes, dichos formularios tendrán un costo estimado de \$ 75.00 (Setenta y cinco dólares netos) cada formulario.

El costo total del sistema es de \$ 1,200 (Mil Doscientos dólares netos) el cual incluye costos indirectos que conllevan a la realización del sistema como los costos de transporte, alimentación, energía eléctrica que se utilizó durante el desarrollo de todos los procesos desde su análisis hasta la etapa de culminación.

Este costo fue obtenido mediante el análisis de la codificación, ya que es un proyecto mediano, el cual no tiene un costo tan elevado.

Análisis de Riesgo

El primer objetivo es evitar el riesgo, pero no es posible evitar todos los riesgos, es por eso que todo el equipo del proyecto, deben de participar en esta actividad de análisis, para desarrollar un plan de contingencias que le permita tomar decisiones de una manera eficaz y contralada que eviten los riesgos.

La identificación del riesgo es un intento sistemático para especificar las amenazas al plan del proyecto (estimaciones, planificación temporal, carga de recursos), identificando los riesgos conocidos y predecibles, el gestor del proyecto da un paso adelante para evitarlos cuando sea posible y contratarlos cuando sea necesario.

Tipos de Riesgo

- **Genéricos:** un método de identificar riesgo es crear una lista de comprobación de elementos de riesgo. Esta lista se puede utilizar para identificar riesgos y se centra en un sub conjunto de riesgos conocidos y predecibles en las siguientes sub categorías.
- **Tamaño del producto:** definir el tiempo de duración por ejemplo el sistema tendrá una duración de 60 a 90 días por lo tanto será de tamaño mediano.
- **Impacto en el Negocio:** en este mundo globalizado en que la tecnología evoluciona día a día, se corre el riesgo que su sistema sea sustituido por nuevos software ya que este proyecto satisface las necesidades actuales de la empresa y es posible que en un futuro la empresa o institución crezca y el sistema requiera nuevas actualizaciones.

- **Características del Cliente:** el buen manejo del sistema depende del grado de preparación y habilidades de cada usuario, también de las exigencias de la empresa o institución.
- **El usuario deberá haber tomado por los menos el curso de operador de microcomputadoras o tener conocimientos mínimos de computación.**
- **Definición del Proceso:** una de las prioridades es diseñar un prototipo de sistema el cual satisfaga las necesidades del cliente, aunque se corre el riesgo de que no se logre cumplir con los objetivos del proyecto.
- **Entorno de Desarrollo:** con respecto a la disponibilidad y calidad de las herramientas que van a emplear en la construcción del producto se debe tener cuidado con lo siguiente:
 - **Incompatibilidad sobre software.**
 - **Falta de proveedores de Licencia de Software.**
 - **Tecnología a Construir:** el sistema está destinado a utilizarse en una intranet, debido a esto puede ocasionar problemas al momento de realizar un ensayo entre el sistema y la intranet, por ejemplo que el servidor no puede visualizar a los clientes.
 - **Tamaño y experiencia de la plantilla:** el equipo de trabajo no cuenta con una amplia experiencia únicamente con los conocimientos teóricos prácticos por lo que no están exentos de problemas en el transcurso del tiempo.

Riesgos Específicos

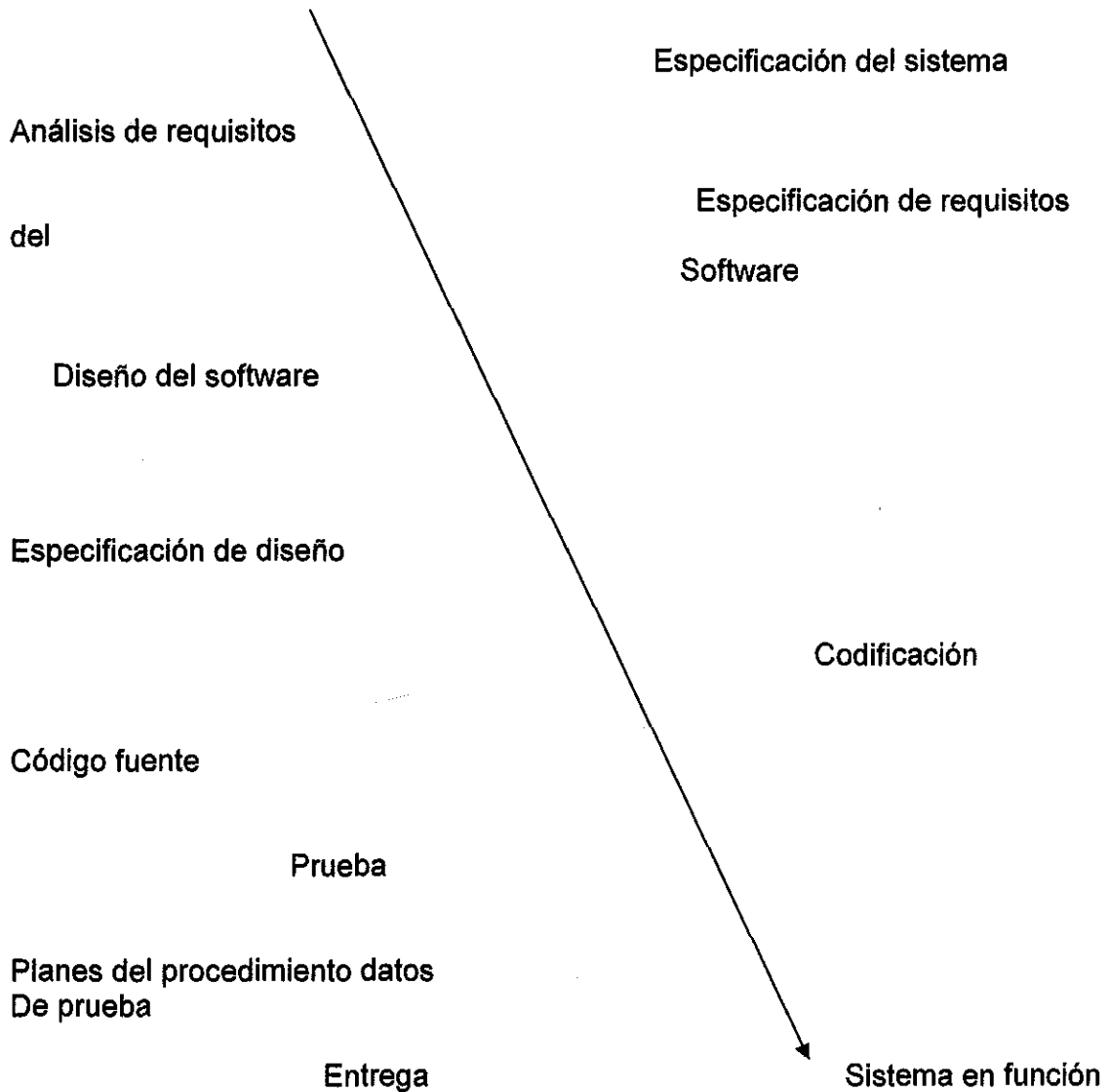
Descripción del Riesgo	Plan de Contingencia
Fallas de energía(voltajes inestables), corto circuitos, cortes de energía	Utilizar equipos de respaldo tales como: baterías, estabilizadores, planta Eléctrica.
Danos o fallos en el equipo de Hardware	Recurrir a los proveedores de los equipos en caso que este vigente la garantía o contratar los servicios de reparación y mantenimiento.
Infección por virus informáticos	Utilizar programas antivirus.
Nuevos requerimientos de sistema	Modificación en costos, planificación y presentar nuevas propuestas.
Daños al software ya sea por virus o problemas de disco duro	Realizar copias actualizadas de respaldo del software a medida que se va desarrollando y tomar medidas de seguridad.
Ausencia de los miembros del equipo por enfermedad, accidentes, etc.	Evaluar si el resto del equipo puede asumir la responsabilidad de seguir con el proyecto o recurrir a la contratación de personal suplente.

Gestión de la Configuración

La gestión de la configuración es un elemento importante de garantía para la calidad del software además del control del cambio, la seguridad del desarrollo adecuado y generación de informes de los cambios realizados.

Una sola línea base es un concepto de lo que es la gestión de configuración del software que controla los cambios sin impedir los cambios injustificados.

Ingeniería del sistema



Requerimientos de la plataforma tecnológica (red)

El Colegio actualmente no cuenta con una plataforma tecnológica LAN y probablemente no habrá crecimiento del personal administrativo, pero el Sistema de Registro desarrollado es un sistema cliente servidor, por tanto se hace necesario de los siguientes equipos de computación:

(1) Estaciones de Trabajo.

Se reemplazará las PC existentes, por 3 PC Nuevas con las siguientes características.

PC's que funcionarán como clientes:

Computador clon, Intel Celeron 4 de 1.6 GHz

- ⊕ Procesador Intel Celeron 4 de 1.6 GHz system bus 800 MHz.
- ⊕ Súper fan cooler original (ventilador) para el procesador.
- ⊕ Mother board o tarjeta madre mercury o pc-chips original
- ⊕ Bus 533
- ⊕ Socket 478
- ⊕ bancos para memoria DDR
- ⊕ Puertos usb 2.0
- ⊕ Puertos ps/2
- ⊕ Puerto serial
- ⊕ Puerto paralelo
- ⊕ Red 10/100 bps
- ⊕ Sonido 3d
- ⊕ Video hasta 128 Mb acelerado
- ⊕ Memoria de 512 Mb DDR pc-2700
- ⊕ Disco duro 80 Gb
- ⊕ Cd-rom negro
- ⊕ Floppy 3 1/2 de 1.44 Mb

- ⊕ Super case atx tower
- ⊕ Teclado ps/2 de 101 teclas español
- ⊕ Mouse 3 botones ps/2 con netscroll
- ⊕ Monitor: SVGA 1024 x 768 resolución (RCT).
Ver anexo ()

(2) Computador que funcionará como Servidor, tendrá las siguientes características:

Computador clon Intel Celeron 4 de 2.8 GHz:

- ⊕ Procesador Intel Celeron 4 de 2.8 GHz system bus 800 MHz.
- ⊕ Súper fan cooler original (ventilador) para el procesador.
- ⊕ Mother board o tarjeta madre mercury o pc-chips original
- ⊕ Bus 533
- ⊕ Socket 775
- ⊕ 2 bancos para memoria DDR
- ⊕ Puertos usb 2.0
- ⊕ Puertos ps/2
- ⊕ Puerto serial
- ⊕ Puerto paralelo
- ⊕ Red 10/100 bps
- ⊕ Sonido 3d
- ⊕ Video hasta 128 Mb acelerado
- ⊕ Memoria de 512 Mb DDR pc-2700
- ⊕ Disco duro 160 Gb
- ⊕ Quemador DVD/CD
- ⊕ Floppy 3 1/2 de 1.44 Mb
- ⊕ Super case atx tower

SISTEMA DE CONTROL DE NOTAS Y MATRICULAS

- ⊕ Teclado ps/2 de 101 teclas español
 - ⊕ Mouse 3 botones ps/2 con netscroll
 - ⊕ Monitor: SVGA 1024 x 768 resolución (RCT).
- Ver anexo ()

Materiales y componentes a utilizar en el desarrollo de la red:

Descripción	Cantidad	Precio Total
Cables	75 mts	\$ 19.50
Conectores de cable	10	\$ 2.5
Switch de 8 Puertos	1	\$ 60.00
Equipos y fuentes especiales	3	\$ 114.00
Canaletas	46 Mts	\$ 220.34
Computadora Intel Celeron	2	\$ 708.00
Capacitación Secretaria	30 horas	\$ 30.00
Codos	3	\$ 3.51
Servidor clon	1	\$ 369.00
Impresora Laser HP 1018	1	\$ 94.00

Características Red Área Local

En cuanto a la seguridad de acceso al Sistema, sólo el Personal autorizado podrá acceder con clave de autenticación, tales como El Director, La Cajera y La Secretaria en horas de trabajo. La secretaria hará uso del Sistema para hacer alguna consulta, registrar información, imprimir historial del alumno.

Se pretende instalar una pequeña LAN en el Colegio con tres computadoras conectadas entre si compartiendo recursos, archivos e impresiones, mediante un Switch. Una de las computadoras cumplirá la función de servidor, pero no será un servidor dedicado, que servirá para el almacenamiento de la información.

Antes de hacer uso del sistema la secretaria y los demás usuario recibirán entrenamiento para comprender el sistema Cliente-Servidor y la pequeña LAN en topología de estrella, teniendo como dispositivo de enlace un Switch de la marca 3Com.

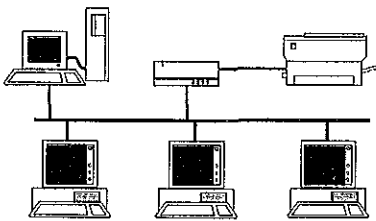
A cada equipo de trabajo se le asignara un nombre, los cables estarán etiquetados debidamente y serán probados antes de realizar la instalación para comprobar que funcionen correctamente y ocurra problemas ya una vez montada la Red.

El local no necesitará hacer ningún cambio, ya que el centro acaba de ser remodelado y cuenta con las condiciones óptimas para el montaje de la red, tampoco habrá necesidad de un Administrador de Red.

Seleccionamos la topología de estrella/bus por las siguientes razones:

Es una configuración combinada. Aquí un multiplexor de señal ocupa la posición del dispositivo central. El sistema de cableado de la red puede tomar la topología bus o anillo. Esto ofrece ventajas en el cableado de edificios que tienen grupos de trabajo separados por distancias considerables.

La topología de estrella es física y la topología de bus es la lógica.



Ventajas de una red con topología estrella bus:

- Es económica.
- El fallo de una estación de trabajo no paraliza al resto de la red.
- Alta expandibilidad, ya que se le pueden agregar muchos nodos sin interrumpir los servicios de otra estación.

La Red es pequeña y no se necesita de monitoreo ni de analizar patrón de tráfico.

Para la red local se utilizaran las siguientes direcciones IP:

- Para el servidor :

IP: 200.62.92.25

Mascara sub red: 255.255.255.0

Puerta de enlace: 200.62.92.1

- Para 1ra. pc cliente :
IP: 200.62.92.26
Mascara sub red: 255.255.255.0
Puerta de enlace: 200.62.92.1
- Para 2da. pc cliente :
IP: 200.62.92.27
Mascara sub red: 255.255.255.0
Puerta de enlace: 200.62.92.1

En cuanto a la información, que diario se haga respaldo de toda la información, datos, Documentos. Y que el resto se mantenga bajo llave el cual tendrá un acceso limitado.

II. Diseño Conceptual

CASOS DE USO

Caso de Uso

En ingeniería del software, un caso de uso es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico.

En otras palabras, un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas.

Caso de Uso: Matricular alumnos	
Objetivo: Registrar todos los alumnos.	
Actor principal: Secretaria	
Personal involucrado e intereses:	
Precondiciones: Registrar Catálogos Secretaria Docente	
Postcondiciones: Generar Reporte	
Flujo básico:	
<u>Acción del actor</u>	<u>Responsabilidad del sistema</u>
Matricular alumnos	Registro de Notas Registrar profesor Guía Registrar Datos del alumno
Flujos alternativos:	
Requisitos especiales:	
Frecuencia: Periodo de Matricula	

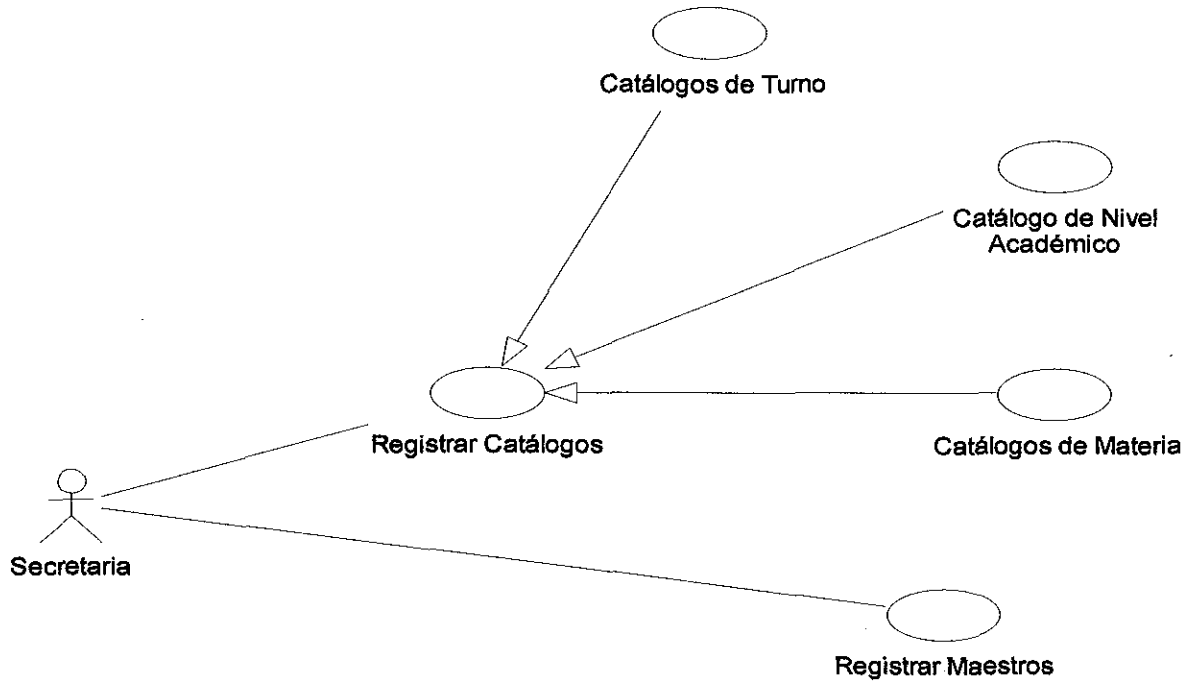
SISTEMA DE CONTROL DE NOTAS Y MATRICULAS

Caso de Uso: Generar reportes	
Objetivo: Mostrar estadísticas de los alumnos matriculados, notas, profesores	
Actor principal: Secretaria	
Personal involucrado e intereses:	
Precondiciones:	
Postcondiciones:	
Flujo básico:	
<u>Acción del actor</u>	<u>Responsabilidad del sistema</u>
Generar Reportes	Generar Reporte de Matriculas Generar Reporte de Notas
Flujos alternativos:	
Requisitos especiales:	
Frecuencia: Inicio de cada mes para reporte de notas Inicio de semestre	

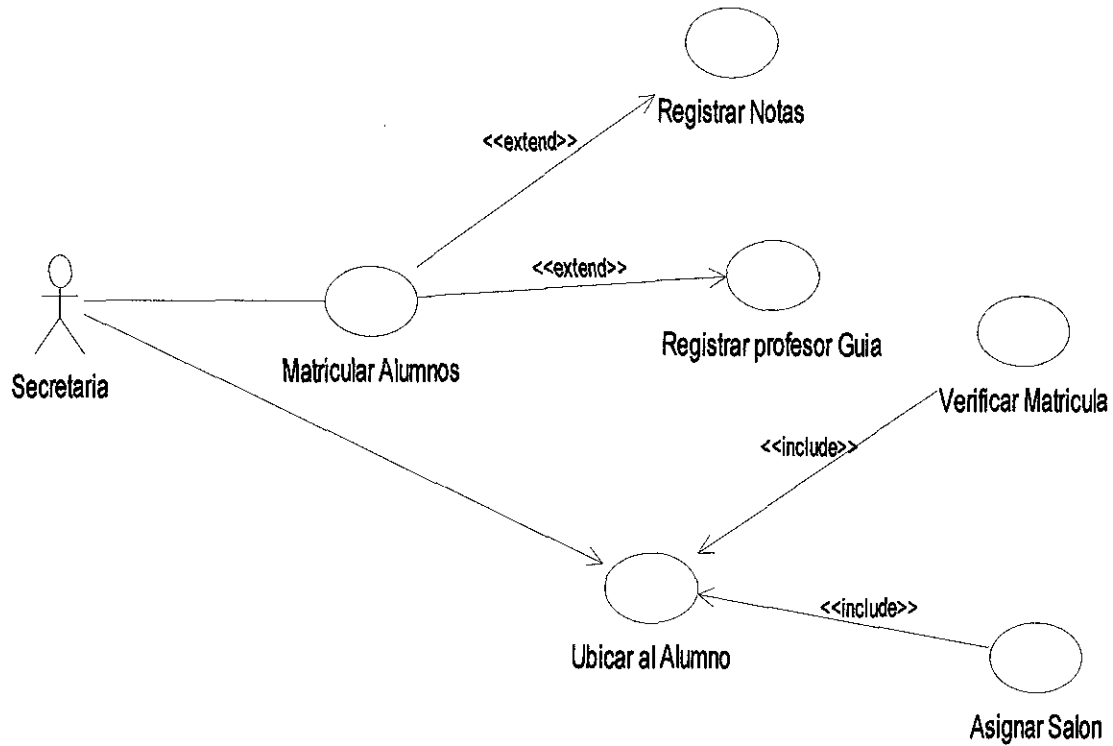
Caso de Uso: Registrar Maestros	
Objetivo: Registrar todos los profesores que imparten clase	
Actor principal: Secretaria	
Personal involucrado e intereses:	
Precondiciones:	
Postcondiciones:	
Flujo básico:	
<u>Acción del actor</u>	<u>Responsabilidad del sistema</u>
Registrar Maestros	Guardar datos del maestro Actualizar datos del maestro Borrar datos del maestro Mostrar datos del maestro
Flujos alternativos:	
Requisitos especiales:	
Frecuencia:	

Caso de Uso: Registrar Catálogos	
Objetivo: Registrar todos los datos de las materias, clases, turno y niveles académicos.	
Actor principal: Secretaria	
Personal involucrado e intereses: Secretaria y alumnos.	
Precondiciones:	
Postcondiciones: Mostrar Alumno	
Flujo básico:	
<u>Acción del actor</u>	<u>Responsabilidad del sistema</u>
Registrar catalogo secretaria docente	Mostrar catálogos secretaria docentes
Registrar catalogo profesores	Catalogo de turno
Flujos alternativos:	
Requisitos especiales:	
Frecuencia: Inicio de semestre	

Caso de Uso Registrar Catálogos



Caso de Uso Matricular Alumno



Caso de Uso Generar Reporte

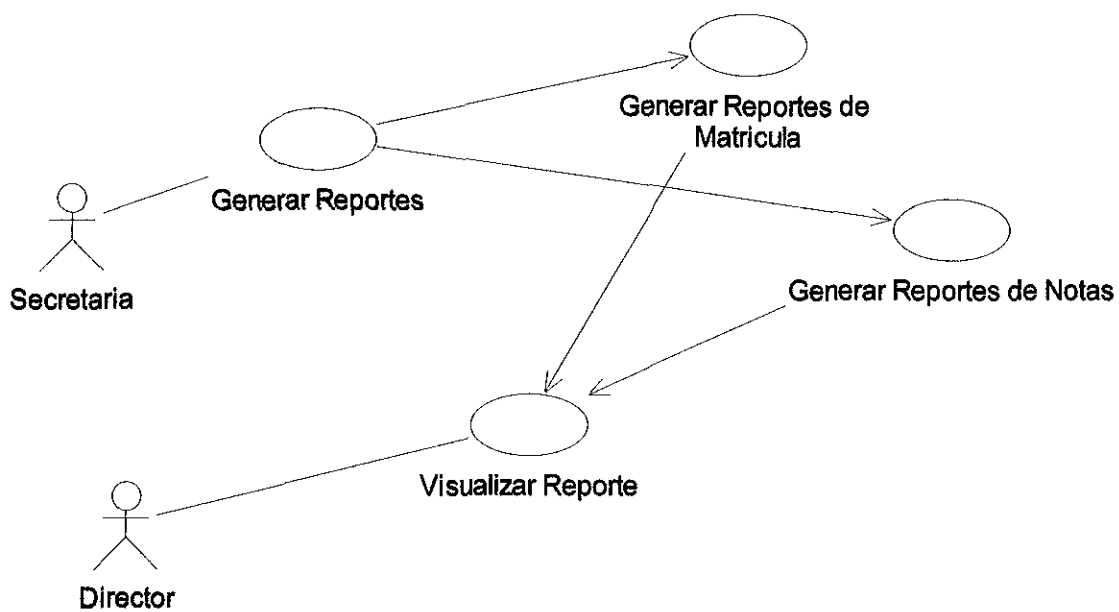


Diagrama de Clase:

En UML, un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas informáticos, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.

Diagrama de Clase Catálogos

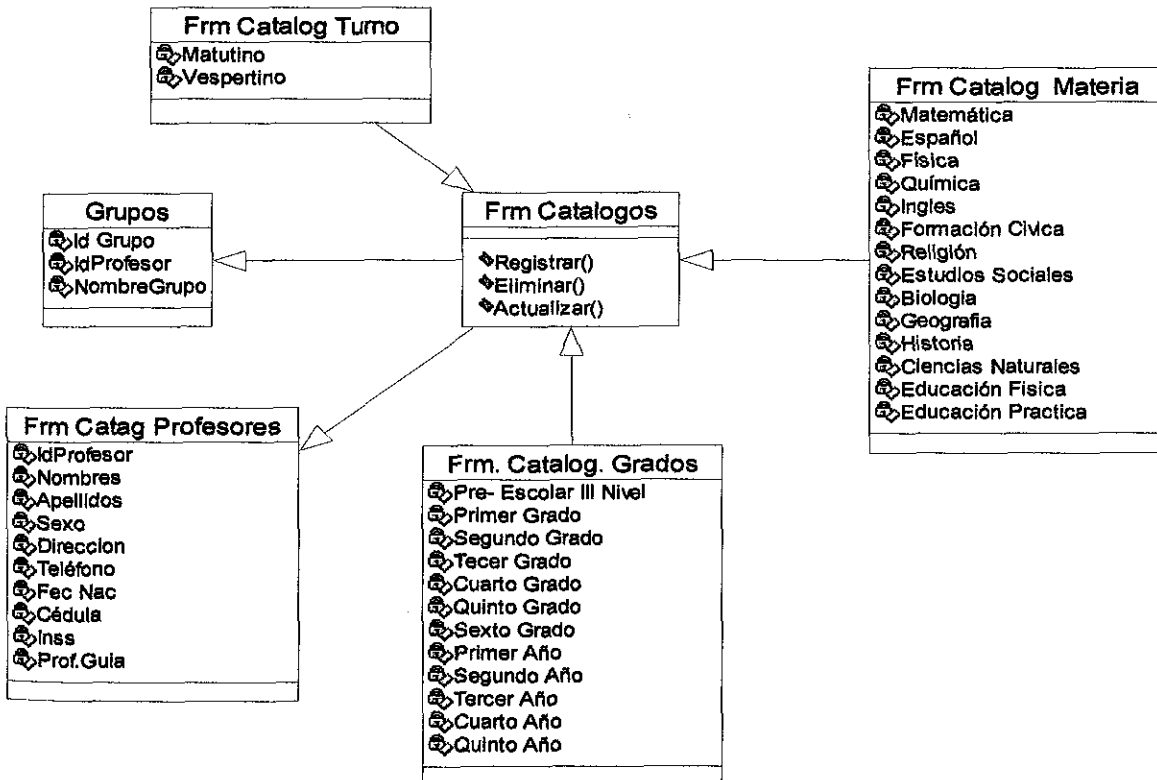


Diagrama de Clase Alumno

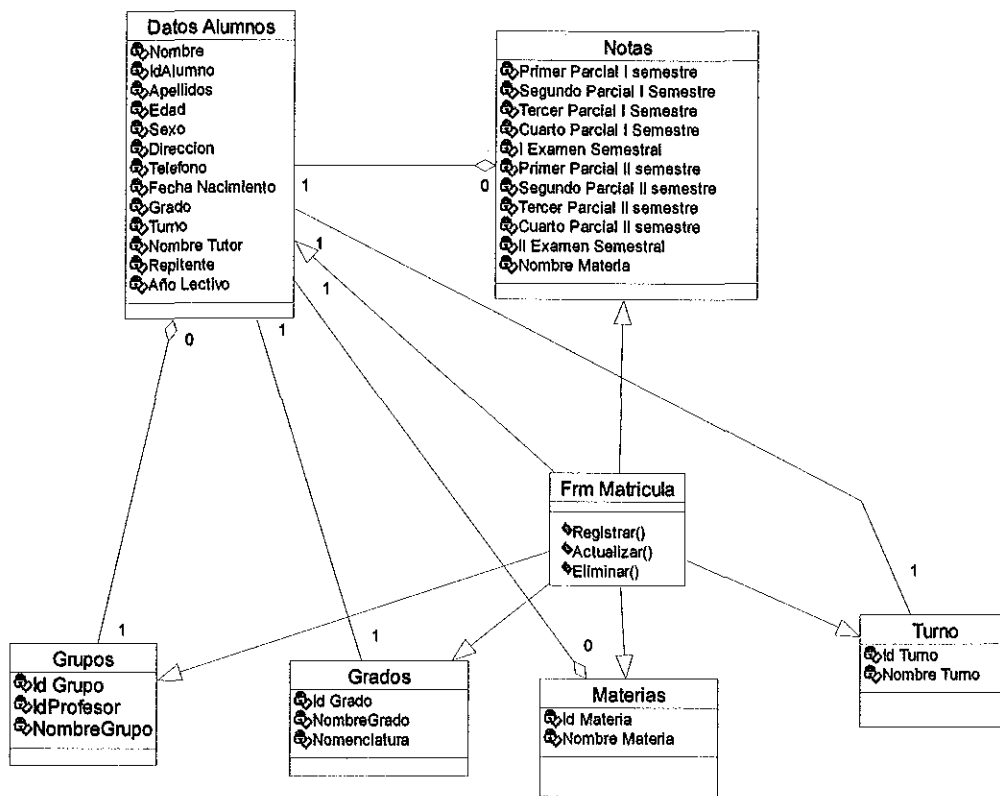


Diagrama de Clase Reporte

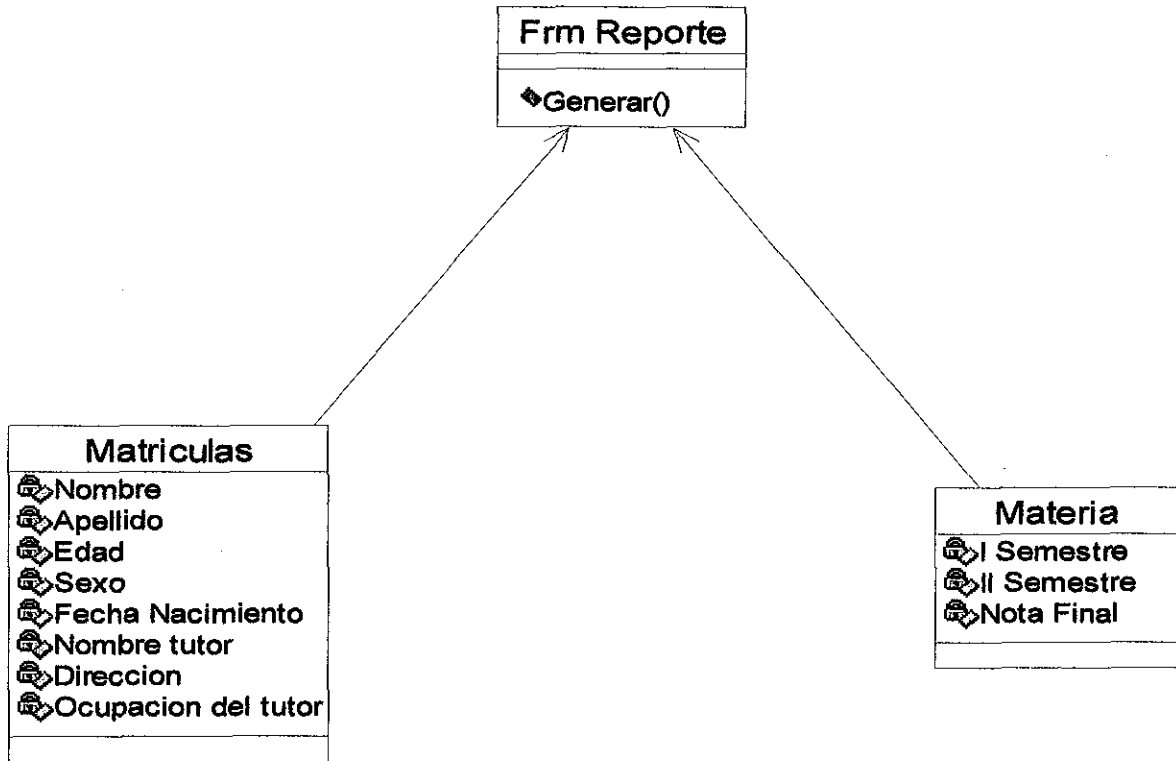


Diagrama de componentes:

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes de Ada, bibliotecas cargadas dinámicamente, etc. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente.

Diagrama de Componentes

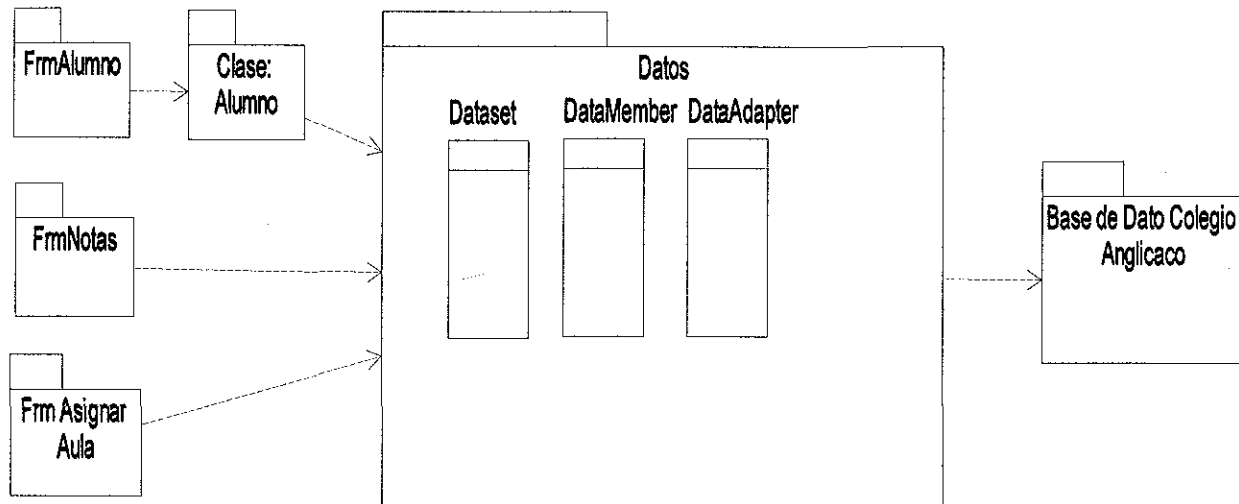


Diagrama de Componentes Catálogos

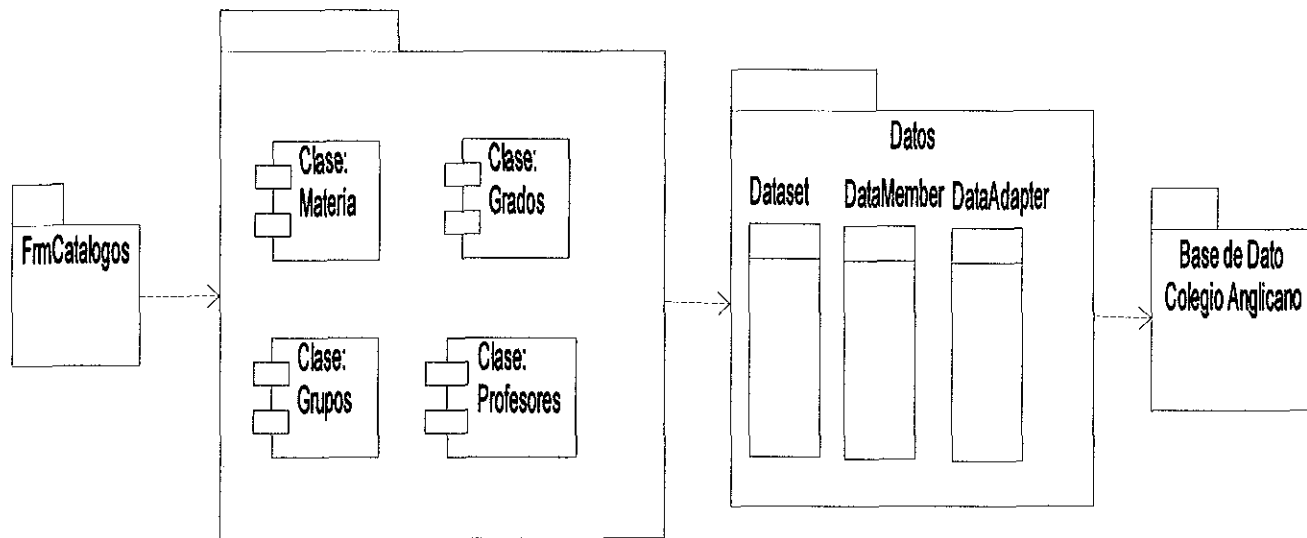


Diagrama de Componentes Reporte

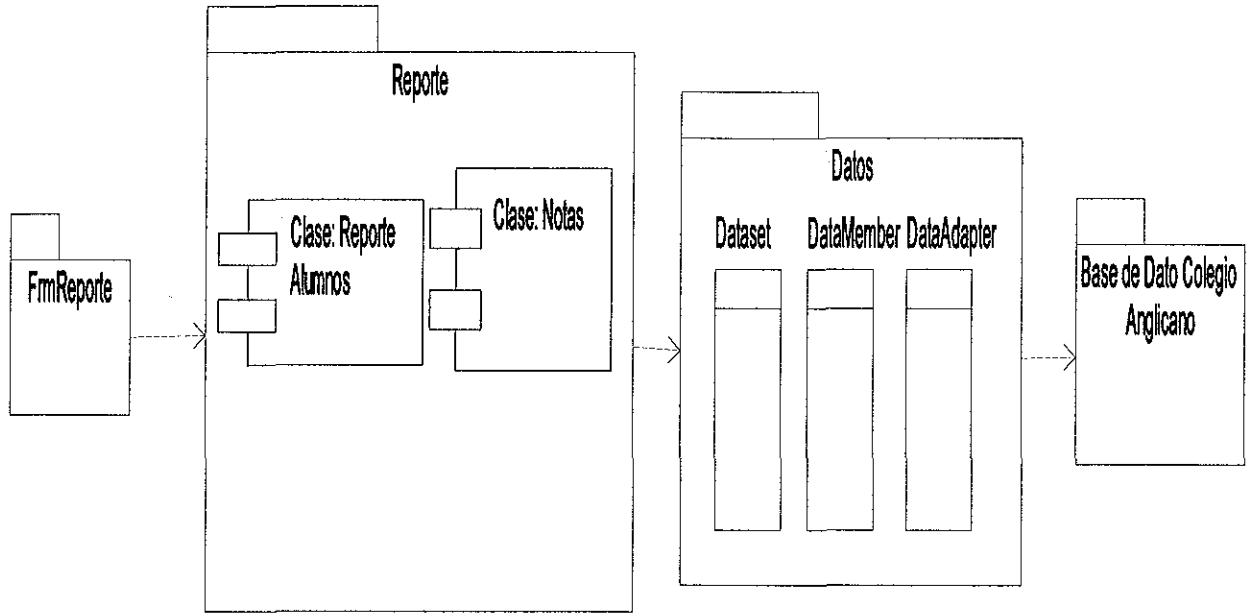


Diagrama de Secuencia:

Un diagrama de secuencia muestra las interacciones entre objetos ordenadas en secuencia temporal. Muestra los objetos que se encuentran en el escenario y la secuencia de mensajes intercambiados entre los objetos para llevar a cabo la funcionalidad descrita por el escenario. En aplicaciones grandes además de los objetos se muestran también los componentes y casos de uso. El mostrar los componentes tiene sentido ya que se trata de objetos reutilizables, en cuanto a los casos de uso hay que recordar que se implementan como objetos cuyo rol es encapsular lo definido en el caso de uso.

Diagrama de Secuencia General

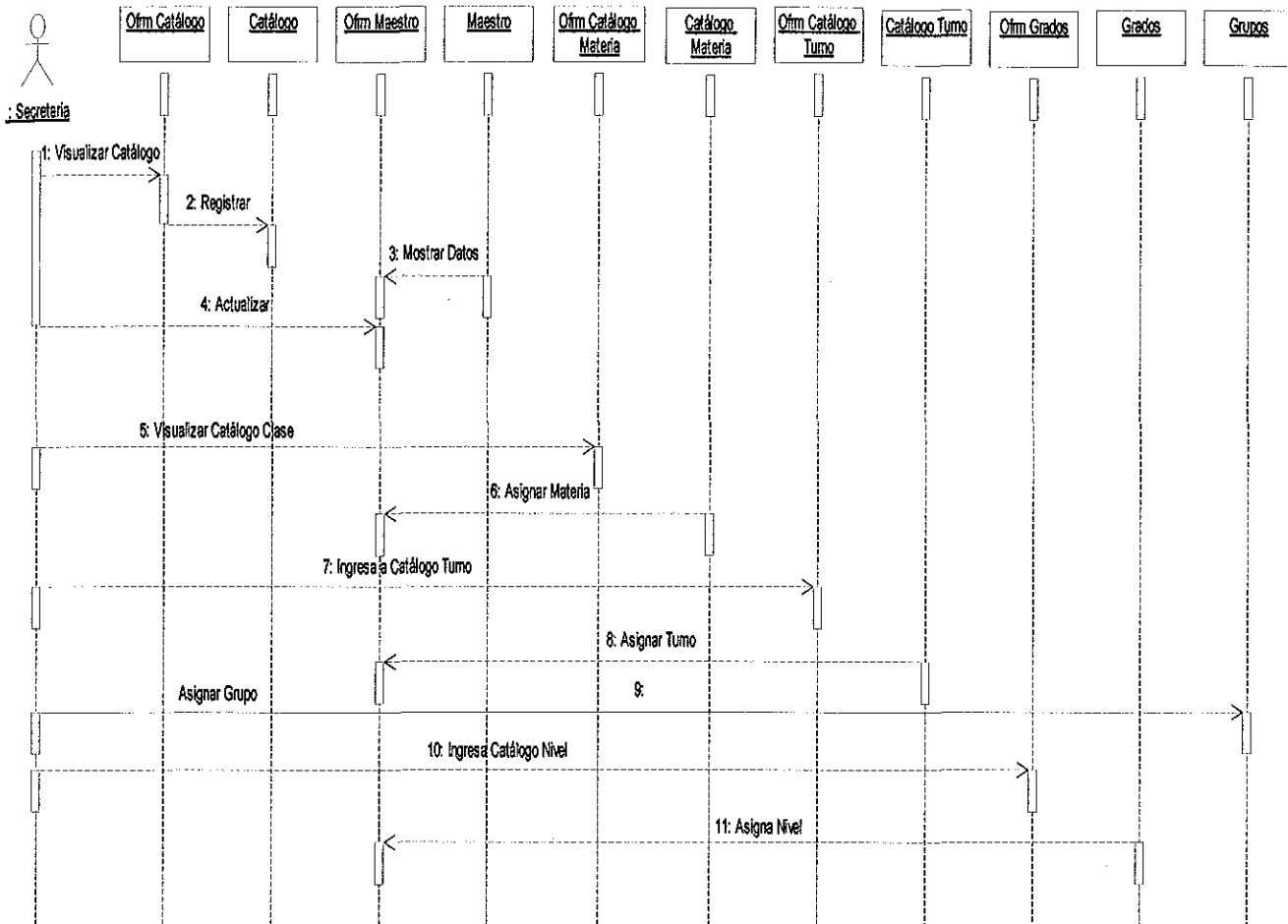


Diagrama de Secuencia Matricula

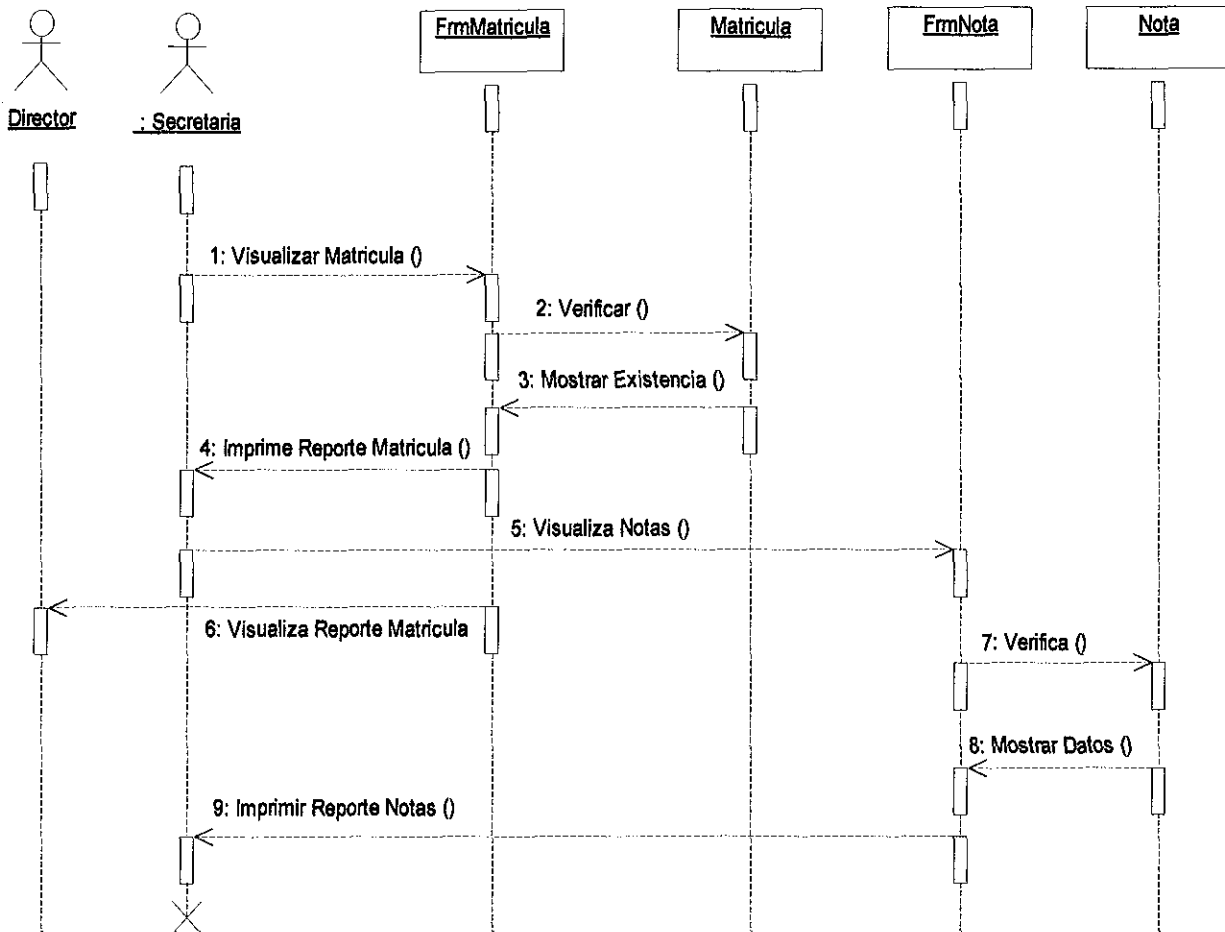


Diagrama de Secuencia Registrar Catalogo

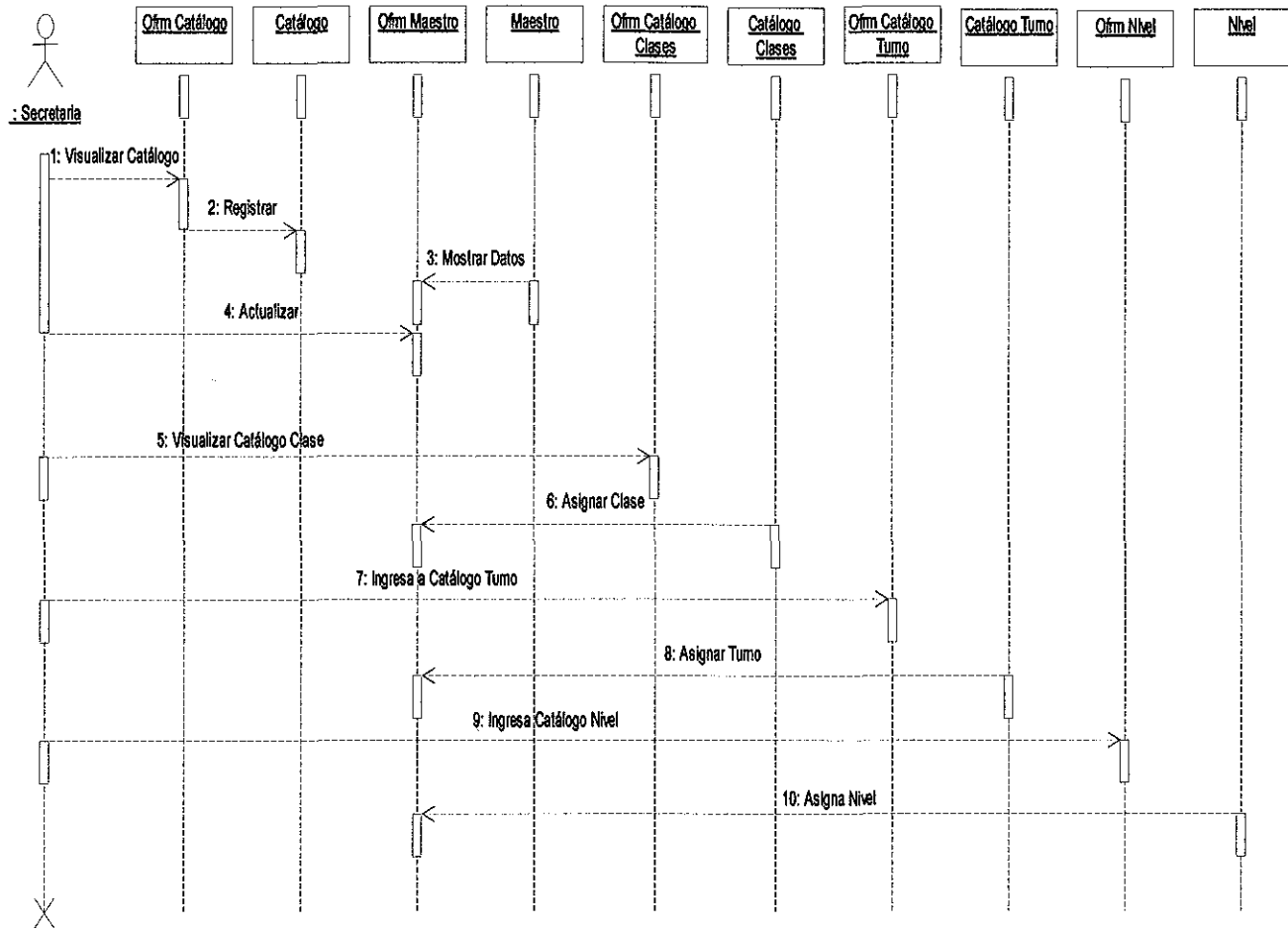


Diagrama de Secuencia Generar Reporte

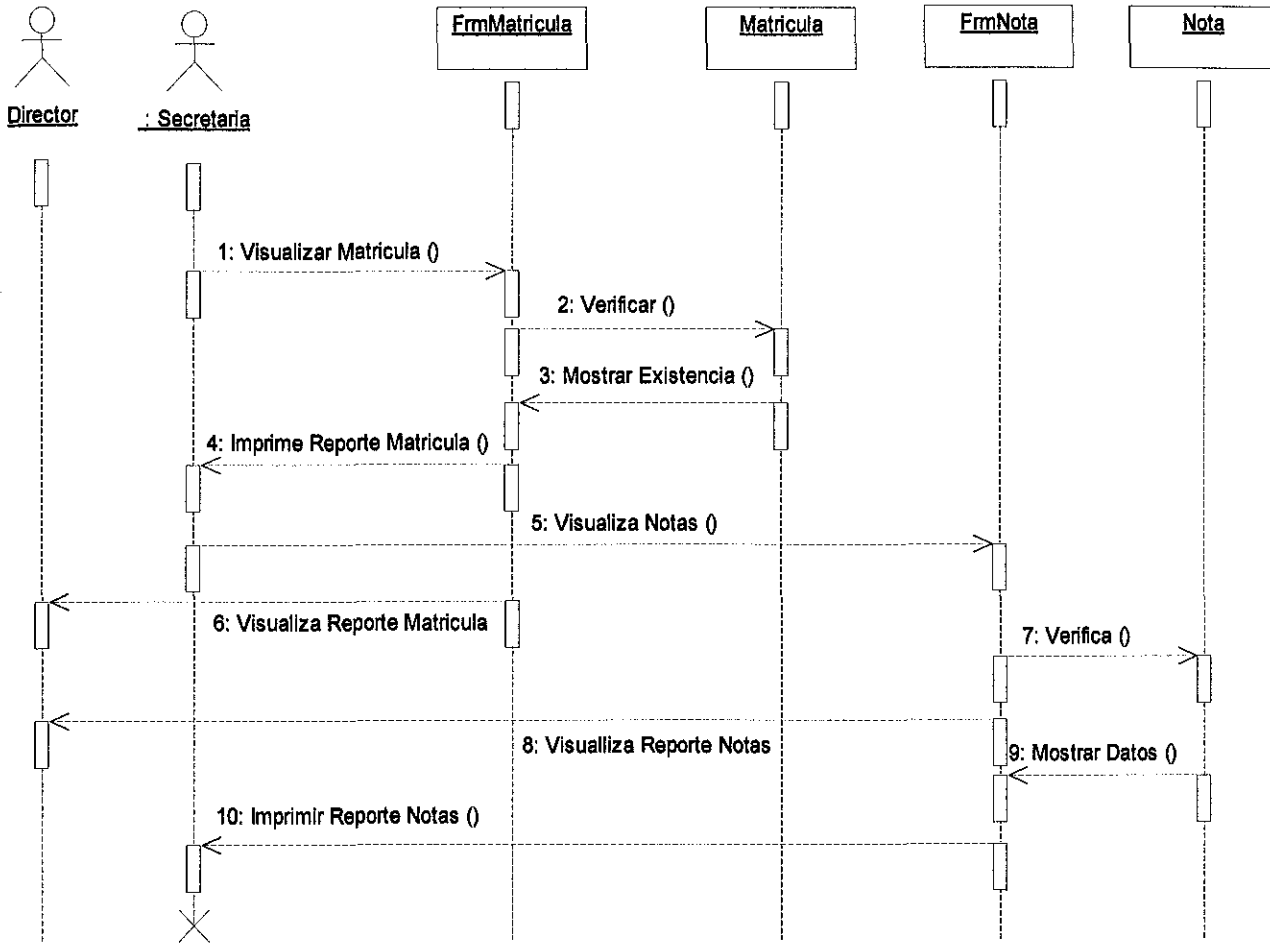


Diagrama de actividad

Los diagramas de actividad describen la secuencia de las actividades en un sistema. Los diagramas de actividad son una forma especial de los diagramas de estado, que únicamente (o mayormente) contienen actividades.

Diagrama de Actividades Registrar Notas

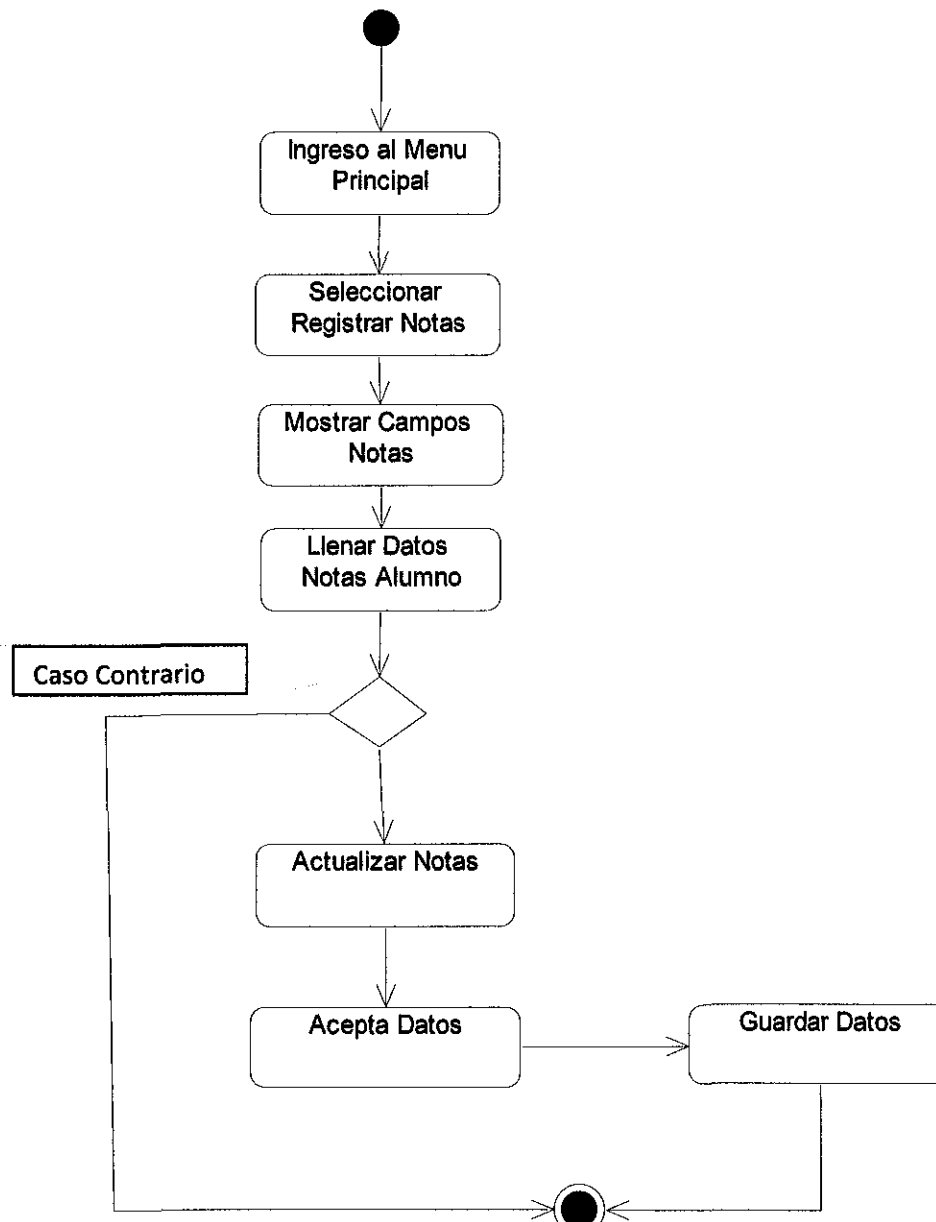


Diagrama de Actividades Generar Reportes

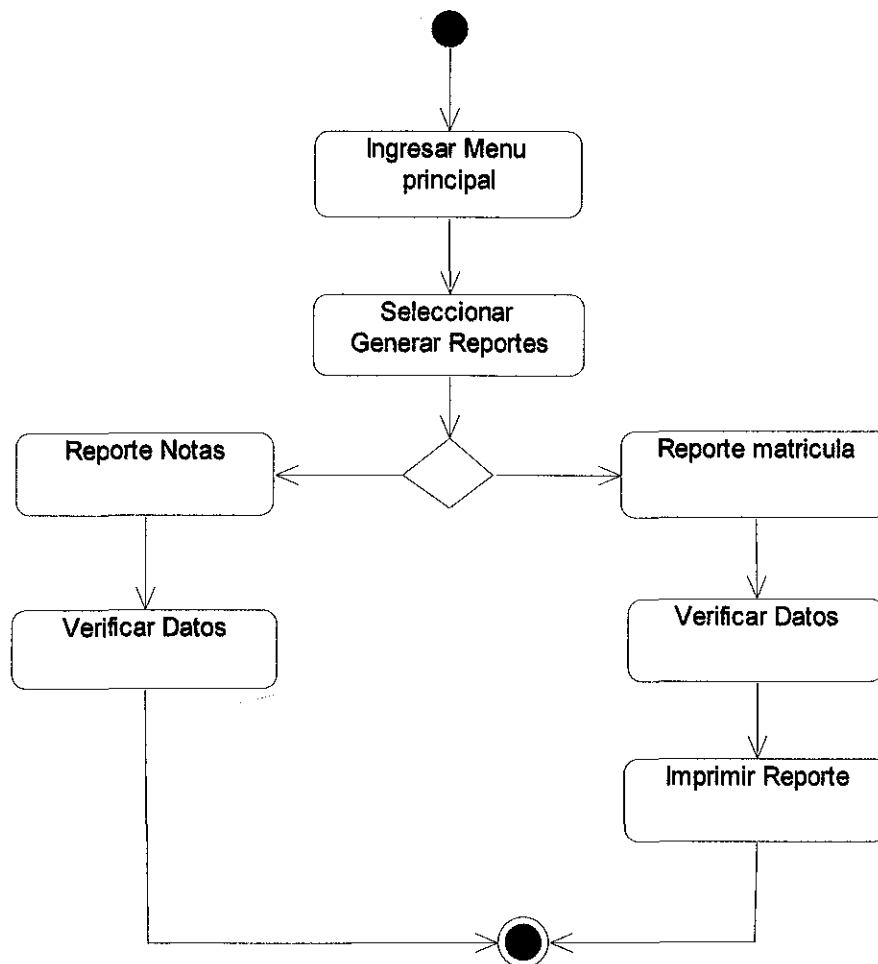


Diagrama de Actividades Registrar Matriculas

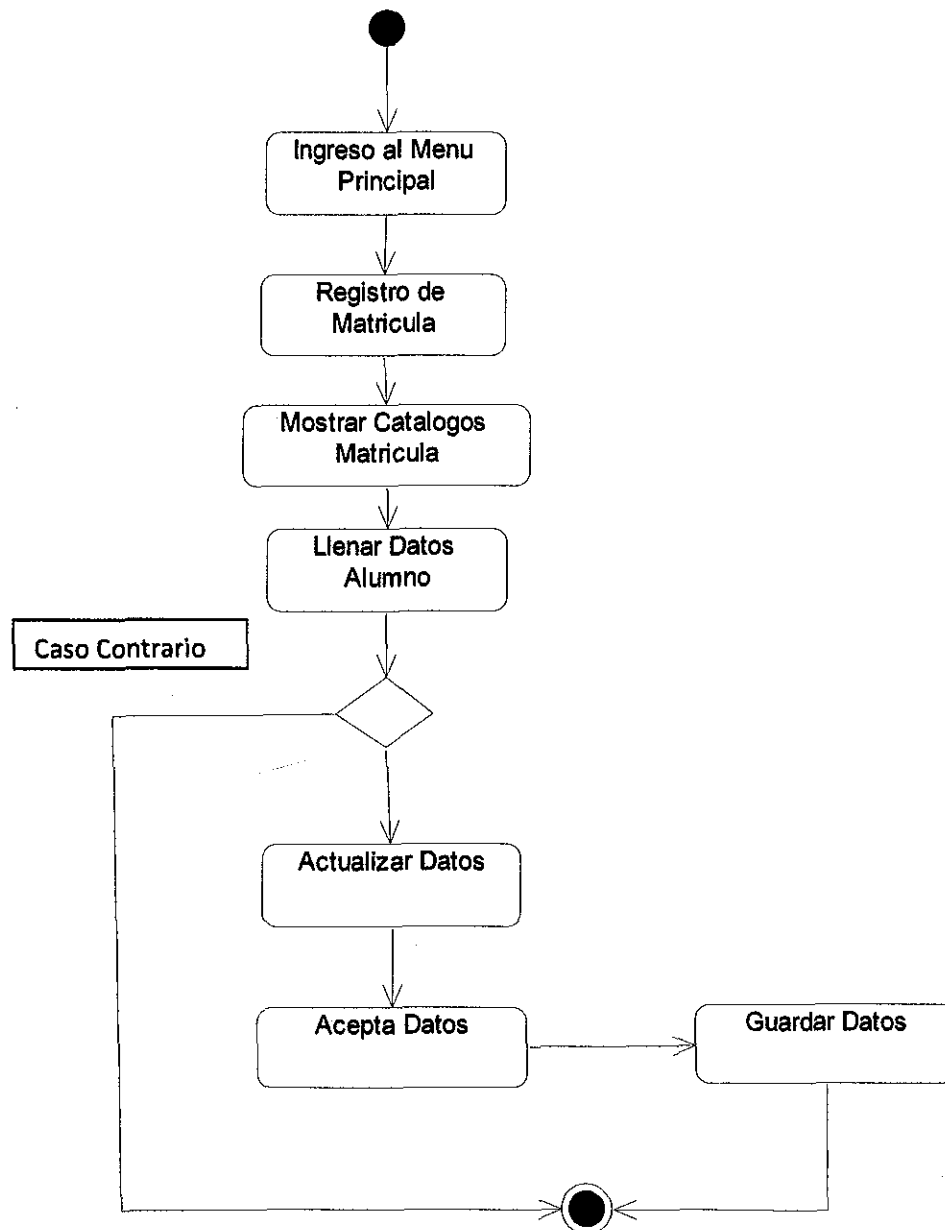
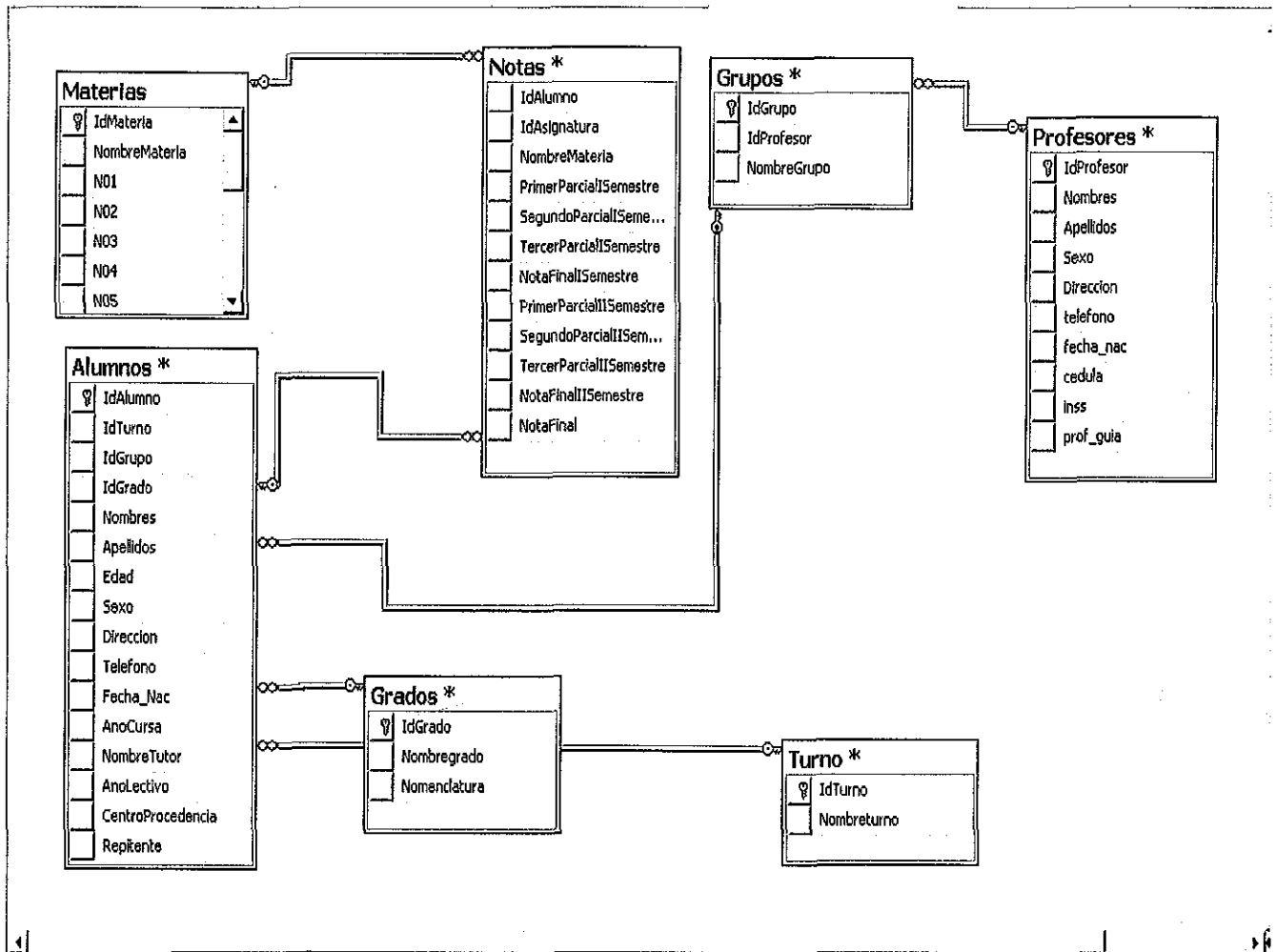


Diagrama Entidad Relación



FASE DE DESARROLLO

I.DISEÑO LOGICO:

Es la conformación de las líneas de código utilizados para la ejecución de cada pantalla en el sistema, así como las tareas y validaciones que ejerce cada uno de los componentes contenidos en los formularios. (Ver anexo No.1).

II.DISEÑO INTERFAZ A NIVEL VISUAL:

Se refiere a la interfaz del sistema, es con la que el usuario interactúa y manipula para realizar todos los procesos requeridos por el mismo y los que el sistema ofrece .son todas las pantallas que el usuario va a visualizar a la hora de ejecutarse el sistema.(Ver anexo No.2).

III. DISEÑO FISICO:

Es donde se muestra el diseño de las tablas de nuestra base de datos, presentándonos los campos que llevan cada una de dichas tablas, el nombre de los campos, tipo del nombre, la longitud de las variables, si permite valores nulos o no los permite y las debidas observaciones en caso de que existan en la tabla. Con el cual se llegaron al diseño del sistema para el colegio anglicano.

Tabla materias:

Nombre de columna	Tipo de datos	Permitir v...
IdMateria	nvarchar(2)	<input type="checkbox"/>
NombreMateria	nvarchar(20)	<input type="checkbox"/>
N01	bit	<input checked="" type="checkbox"/>
N02	bit	<input checked="" type="checkbox"/>
N03	bit	<input checked="" type="checkbox"/>
N04	bit	<input checked="" type="checkbox"/>
N05	bit	<input checked="" type="checkbox"/>
N06	bit	<input checked="" type="checkbox"/>
N07	bit	<input checked="" type="checkbox"/>
N08	bit	<input checked="" type="checkbox"/>
N09	bit	<input checked="" type="checkbox"/>
N10	bit	<input checked="" type="checkbox"/>
N11	bit	<input checked="" type="checkbox"/>
N12	bit	<input checked="" type="checkbox"/>
N13	bit	<input checked="" type="checkbox"/>
N14	bit	<input checked="" type="checkbox"/>

Tabla notas:

Nombre de columna	Tipo de datos	Permitir valores nulos
IdAlumno	nvarchar(4)	<input checked="" type="checkbox"/>
IdAsignatura	nvarchar(2)	<input checked="" type="checkbox"/>
NombreMateria	nvarchar(20)	<input checked="" type="checkbox"/>
PrimerParcialISemestre	float	<input type="checkbox"/>
SegundoParcialISemes...	float	<input type="checkbox"/>
TercerParcialISemestre	float	<input type="checkbox"/>
NotaFinalISemestre	float	<input type="checkbox"/>
PrimerParcialIISemestre	float	<input type="checkbox"/>
SegundoParcialIISeme...	float	<input type="checkbox"/>
TercerParcialIISemestre	float	<input type="checkbox"/>
NotaFinalIISemestre	float	<input type="checkbox"/>
NotaFinal	float	<input type="checkbox"/>

Tabla Profesores:


Nombre de columna	Tipo de datos	Permitir valores nulos
 IdProfesor	nvarchar(2)	<input type="checkbox"/>
Nombres	nvarchar(50)	<input type="checkbox"/>
Apellidos	nvarchar(50)	<input type="checkbox"/>
Sexo	char(1)	<input type="checkbox"/>
Direccion	nvarchar(50)	<input checked="" type="checkbox"/>
telefono	nvarchar(20)	<input checked="" type="checkbox"/>
fecha_nac	datetime	<input type="checkbox"/>
cedula	nvarchar(16)	<input type="checkbox"/>
inss	nvarchar(12)	<input checked="" type="checkbox"/>
prof_guia	bit	<input type="checkbox"/>

Tabla de turno:

Nombre de columna	Tipo de datos	Permitir valores nulos
IdTurno	nvarchar(2)	<input type="checkbox"/>
Nombretorno	nvarchar(20)	<input type="checkbox"/>

Tabla alumno:

Nombre de columna	Tipo de datos	Permitir valores nulos
IdAlumno	nvarchar(4)	<input type="checkbox"/>
IdTurno	nvarchar(2)	<input type="checkbox"/>
IdGrupo	nvarchar(2)	<input type="checkbox"/>
Nombres	nvarchar(40)	<input type="checkbox"/>
Apellidos	nvarchar(50)	<input type="checkbox"/>
Edad	int	<input type="checkbox"/>
Sexo	char(1)	<input type="checkbox"/>
Direccion	nvarchar(50)	<input checked="" type="checkbox"/>
Telefono	nvarchar(10)	<input checked="" type="checkbox"/>
Fecha_Nac	datetime	<input type="checkbox"/>
AnoCursa	nvarchar(2)	<input checked="" type="checkbox"/>
NombreTutor	nvarchar(50)	<input checked="" type="checkbox"/>
AnoLectivo	int	<input type="checkbox"/>
CentroProcedencia	nchar(30)	<input checked="" type="checkbox"/>
Repitente	bit	<input checked="" type="checkbox"/>

Tabla grados:

	Nombre de columna	Tipo de datos	Permitir valores nulos
?	IdGrado	nvarchar(2)	<input type="checkbox"/>
	Nombregrado	nvarchar(20)	<input type="checkbox"/>
	Nomenclatura	char(3)	<input checked="" type="checkbox"/>

Tabla grupos:

	Nombre de columna	Tipo de datos	Permitir valores nulos
?	IdGrupo	nvarchar(2)	<input type="checkbox"/>
	IdProfesor	nvarchar(2)	<input type="checkbox"/>
	NombreGrupo	nvarchar(20)	<input type="checkbox"/>

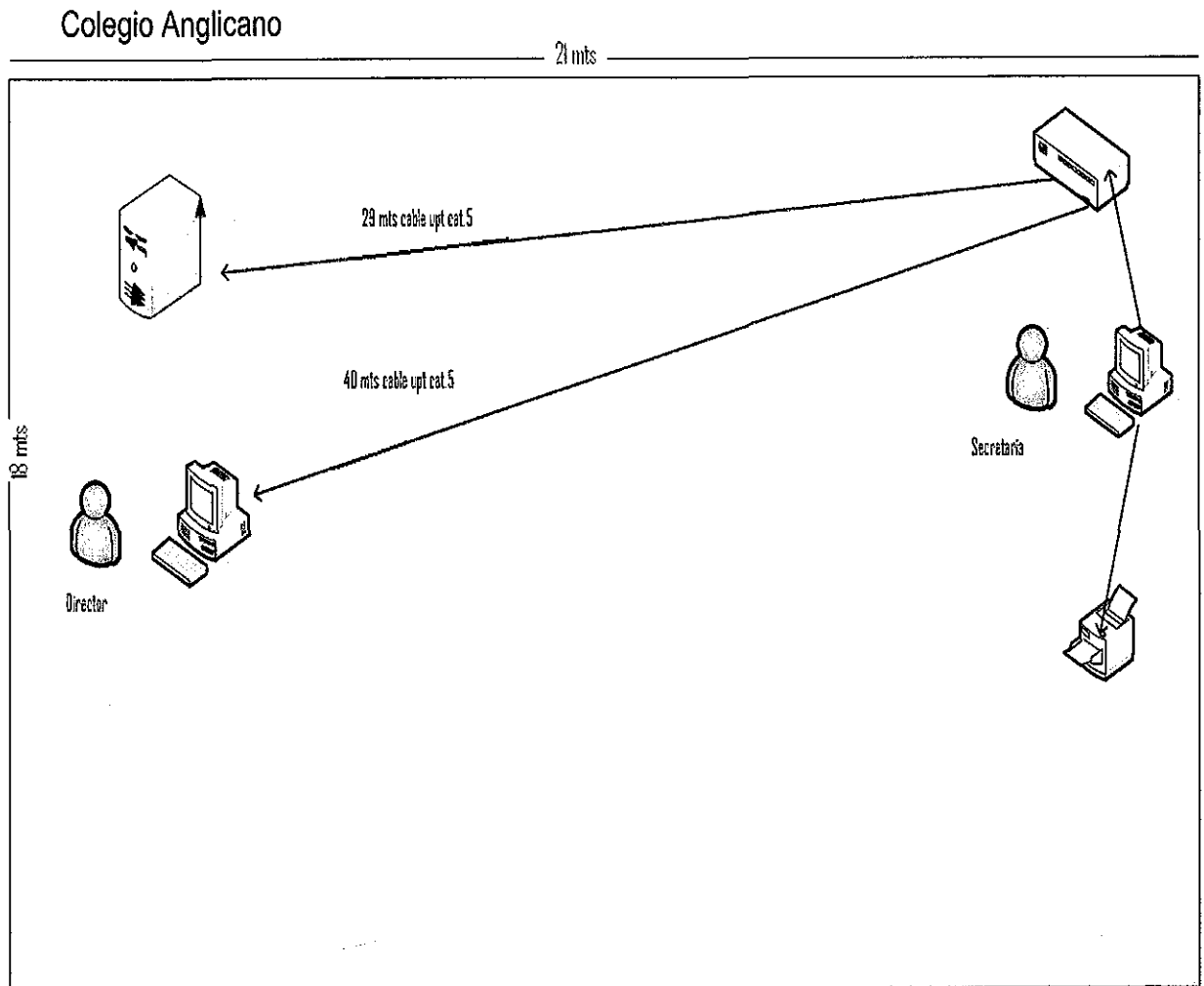
Red de Área Local

Etapa de Realización:

El local donde se desarrollará la red cuenta con las siguientes características:

- Energía eléctrica con polo a tierra.
- Cielo falso.
- Cuenta con el mobiliario de oficina.
- Cuenta con dos puertas de acceso.
- El ancho de dicho local es 21 mts.
- El largo de dicho local es 18 mts.
- El alto de dicho de local es 4 mts.
- El Switch será ubicado en la oficina de la secretaria, en dicha oficina se encontrara una de las pc.
- En la oficina del director se encontrara otra de las pc.
- El servidor será ubicado en una oficina que cuenta con los requisitos para alojar una pc.
- La distancia del switch hacia la pc de la secretaria es de 4 mts, pero el cable tendrá un tamaño de 6 metros.
- La distancia del switch hacia el servidor es de 21 mts y del techo hacia el suelo la distancia es de 4 mts, pero el cable tendrá un tamaño de 29 metros.
- La distancia del switch hacia la pc ubicada en la oficina del director es de 34 mts y del techo hacia el suelo la distancia es de 4 mts, pero el cable tendrá un tamaño de 40 metros.
- Se utilizaran 46 mts de canaleta.
- Se utilizarán 3 codos.

Diagrama Red Área Local



CONCLUSIONES

Tomando en cuenta la constante evolución de la tecnología en este mundo globalizado, sabemos que la informática tiene un papel importante dentro del desarrollo de las sociedades.

Por lo cual, se concluye que al implementar un modelo tecnológico en el colegio anglicano de puerto cabezas (R.A.A.N) se logrará el primer paso para la integración de una sociedad al mundo digital, que hasta ahora se ha mantenido al margen.

Se han alcanzado objetivos propuestos al desarrollar una herramienta que permita el cambio en la administración de la información del colegio anglicano en el manejo de sus procesos, para el registro de matriculas y notas. Fortaleciendo el control de la información, que es el principal activo de toda empresa o institución para la realizar sus tareas y la generación de reportes requeridos, cumpliendo con éxito con los objetivos al final del desarrollo del software de aplicación.

RECOMENDACIONES

Habiendo llegado a la conclusión del desarrollo del software de aplicación, se deben dar ciertas recomendaciones para el buen funcionamiento del sistema posterior a la implementación en el colegio anglicano las cuales son:

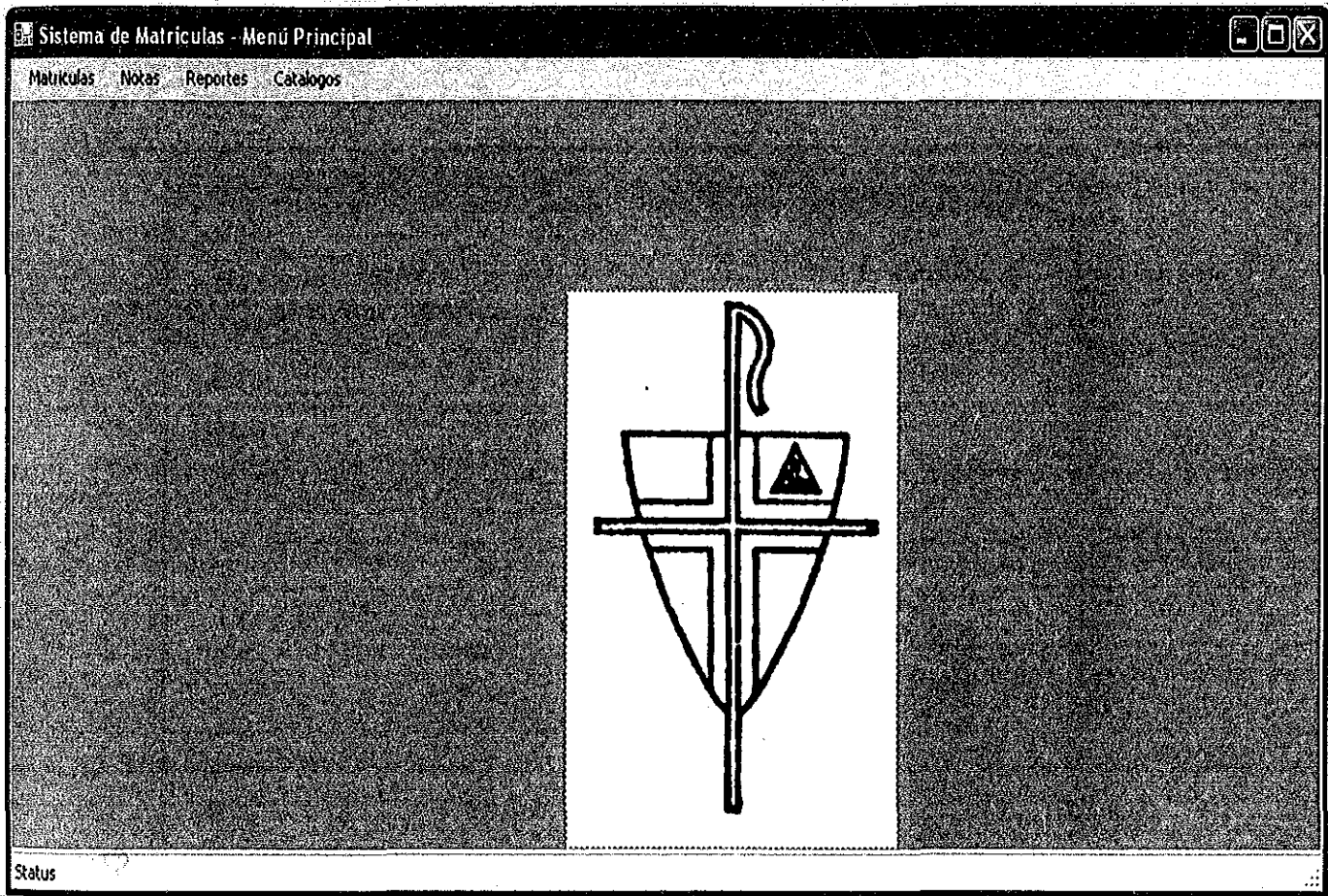
- ❖ Mantenimiento preventivo de los equipos por lo menos una vez al mes.
- ❖ Respaldo de la información que se encuentra en el servidor en discos ópticos para mayor seguridad de los datos.
- ❖ Evitar el manejo no autorizado de los equipos y del software instalado.
- ❖ Evitar el daño o remover físicamente la red LAN instalada.
- ❖ Recurrir al manual de usuario siempre que exista una duda, respecto al software desarrollado.
- ❖ En caso de que se presente un problema mayor y no se pueda resolver internamente dentro del centro por la secretaria, ponerse en contacto con los desarrolladores de la aplicación.

ANEXOS

Anexo 1

DISEÑO LOGICO (CODIGOS)

MENU PRINCIPAL



EN LA BARRA DE HERRAMIENTAS DEL MENU PRINCIPAL EN MATRICULAS

CODIGO PARA CARGAR EL FORMULARIO REGISTRAR MATRICULAS EN EL MENU PRINCIPAL

```
Private Sub AlumnosToolStripMenuItem_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
AlumnosToolStripMenuItem.Click
```

```
    FrmAlumnos.ShowDialog()
```

```
    FrmAlumnos.Close()
```

```
End Sub
```

The screenshot shows a Windows application window titled "Registro de Alumnos - Matricula". The window contains a registration form with the following fields and controls:

- Id Alumno: Text box
- Nombres: Text box
- Apellidos: Text box
- Direccion: Text box
- Fecha_Nac: Date picker (format: / /)
- Edad: Text box
- Sexo: Radio buttons
- Telefono: Text box
- Nombre Tutor: Text box
- Año Lectivo: Dropdown menu (AxTDBCo)
- C.Procedencia: Text box
- Repitente: Check box
- Grado: Dropdown menu (AxTDBCombo3)
- Turno: Dropdown menu (AxTDBCombo1)
- CLASES QUE LLEVARA EL ALUMNO: Section header
- Label14, Label16, Label17, Label18, Label19: Labels on the left
- Label20, Label21, Label22, Label23, Label24: Labels on the right
- Adodc1, Adodc2, Adodc3: Data adapters with navigation buttons
- Barra de Botones: A toolbar with various navigation and action buttons
- Imprimir: Print button
- Label26: Label at the bottom left

CODIGO FORMULARIO REGISTRAR MATRICULAS

FORM ALUMNOS

```
Imports System.Data
```

```
Imports System.Data.SqlClient
```

```
Public Class FrmAlumnos
```

```
    Dim FILAS() As DataRow
```

```
    Dim odatarow As DataRow
```

```
    Dim odatarow1 As DataRow
```

```
    Private ipos As Integer = 0, ipos1 As Integer = 0
```

```
    Dim oDataSet As DataSet
```

```
    Dim oAdaptador As SqlDataAdapter
```

```
    Dim agregando As Boolean, editando As Boolean, undo As Boolean
```

```
    Dim oCmdBuilder As SqlCommandBuilder
```

```
    Dim oComando As SqlCommand
```

```
    Dim oConexion As New SqlConnection
```

```
    Dim XXX As String, num As String, xcad As String, y As Integer, i  
As Integer
```

```
    Private Sub FrmAlumnos_Deactivate(ByVal sender As Object, ByVal e  
As System.EventArgs) Handles Me.Deactivate
```

```
        oConexion.Close()
```

```
    End Sub
```

```
    Private Sub FrmAlumnos_Load(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles MyBase.Load
```

```
        oConexion = New SqlConnection("Server=(local);" & _
```

```
        "DataBase=colegioanglicano;" & _
```

```

"Integrated Security=SSPI;Connect Timeout=5")

oComando = New SqlCommand

oComando.Connection = oConexion

oComando.CommandText = "SELECT * FROM Alumnos"

oAdaptador = New SqlDataAdapter(oComando)

oCmdBuilder = New SqlCommandBuilder(oAdaptador)

oDataSet = New DataSet

oConexion.Open()

oAdaptador.Fill(oDataSet, "Alumnos")

If oDataSet.Tables("Alumnos").Rows.Count = 0 Then

    Me.Label14.Text = ""

    Me.Label16.Text = ""

    Me.Label17.Text = ""

    Me.Label18.Text = ""

    Me.Label19.Text = ""

    Me.Label20.Text = ""

    Me.Label21.Text = ""

    Me.Label22.Text = ""

    Me.Label23.Text = ""

    Me.Label24.Text = ""

Else

    odatarow = oDataSet.Tables("ALUMNOS").Rows(ipos)

    Me.TextBox1.Text = odatarow("IDALUMNO")

    Me.TextBox2.Text = odatarow("Nombres")

    Me.TextBox3.Text = odatarow("apellidos")

    Me.TextBox6.Text = odatarow("direccion")

```

```

Me.MaskedTextBox1.Text = odatarow("fecha_nac")
Me.TextBox5.Text = odatarow("edad")
Me.TextBox8.Text = odatarow("sexo")
Me.TextBox7.Text = odatarow("telefono")
Me.TextBox10.Text = odatarow("Nombretutor")
Me.TextBox4.Text = odatarow("centroprocedencia")
If odatarow("repite") Then
    Me.CheckBox1.CheckState = True
Else
    Me.CheckBox1.CheckState = False
End If

Me.AxTDBCombo1.BoundText = odatarow("idturno")
Me.AxTDBCombo2.BoundText = odatarow("anolectivo")
Me.AxTDBCombo3.BoundText = odatarow("anocursa")

Me.Label26.Text = "Recs. " & Me.ipos + 1 & " de " &
oDataSet.Tables("Alumnos").Rows.Count

mostrar_materias(odatarow("anocursa"))

End If

```

```

Me.ButtonGuardar.Enabled = False
Me.ButtonInicio.Enabled = True
Me.ButtonAnterior.Enabled = True
Me.ButtonNuevo.Enabled = True
Me.ButtonBorrar.Enabled = True
Me.ButtonSiguiente.Enabled = True

```

```
Me.ButtonUltimo.Enabled = True
Me.ButtonEditor.Enabled = True
Me.ButtonDeshacer.Enabled = False

Me.TextBox1.ReadOnly = True
Me.TextBox2.ReadOnly = True
Me.TextBox3.ReadOnly = True
Me.TextBox6.ReadOnly = True
Me.MaskedTextBox1.ReadOnly = True
Me.TextBox5.ReadOnly = True
Me.TextBox8.ReadOnly = True
Me.TextBox7.ReadOnly = True
Me.TextBox10.ReadOnly = True
Me.TextBox4.ReadOnly = True
Me.CheckBox1.Enabled = False
Me.AxTDBCombo1.Enabled = False
Me.AxTDBCombo2.Enabled = False
Me.AxTDBCombo2.Enabled = False
agregando = False
editando = False
```

End Sub

BOTON BORRAR EN FORMULARIO ALUMNOS

```
Private Sub ButtonBorrar_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ButtonBorrar.Click
```

```
Dim borrar As String
```

```
borrar = MsgBox("Desea eliminar el registro", MsgBoxStyle.YesNo  
+ MsgBoxStyle.Question, "Advertencia")
```

```
If borrar = MsgBoxResult.Yes Then
```

```
Dim eliminar As New ClassAlumno
```

```
eliminar.borrar_Alumnos(ipos)
```

```
refrescar()
```

```
End If
```

```
End Sub
```

```
Private Sub refrescar()
```

```
oConexion = New SqlConnection("Server=(local)\SQLEXPRESS;" & _
```

```
"DataBase=colegioanglicano;" & _
```

```
"Integrated Security=SSPI;Connect Timeout=5")
```

```
oComando = New SqlCommand
```

```
oComando.Connection = oConexion
```

```
oComando.CommandText = "SELECT * FROM Alumnos"
```

```
oAdaptador = New SqlDataAdapter(oComando)
```

```
oCmdBuilder = New SqlCommandBuilder(oAdaptador)
```

```
oDataSet = New DataSet
```

```
oConexion.Open()
```

```
oAdaptador.Fill(oDataSet, "Alumnos")
```

```
'ipos = 0
```

```
odatarow = oDataSet.Tables("Alumnos").Rows(ipos)
```

```
Me.muestra_datos()
```

```
    mostrar_materias(odatarow("anocursa"))

    Me.Label26.Text = "Recs. " & Me.ipos + 1 & " de " &
oDataSet.Tables("Alumnos").Rows.Count

End Sub
```

```
Public Sub muestra_datos()

    odatarow = oDataSet.Tables("Alumnos").Rows(ipos)

    Me.TextBox1.Text = odatarow("idalumno")
    Me.TextBox2.Text = odatarow("Nombres")
    Me.TextBox3.Text = odatarow("apellidos")
    Me.TextBox6.Text = odatarow("direccion")
    Me.MaskedTextBox1.Text = odatarow("fecha_nac")
    Me.TextBox5.Text = odatarow("edad")
    Me.TextBox8.Text = odatarow("sexo")
    Me.TextBox7.Text = odatarow("telefono")
    Me.TextBox10.Text = odatarow("Nombretutor")
    Me.TextBox4.Text = odatarow("centroprocedencia")

    If odatarow("repitente") Then

        Me.CheckBox1.CheckState = True

    Else

        Me.CheckBox1.CheckState = False

    End If

    Me.AxTDBCombo1.BoundText = odatarow("idturno")
    Me.AxTDBCombo2.BoundText = odatarow("anolectivo")
    Me.AxTDBCombo3.BoundText = odatarow("anocursa")

End Sub
```

```
Private Sub GroupBox1_Enter(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles GroupBox1.Enter
```

```
End Sub
```

```
Private Sub TextBox5_GotFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles TextBox5.GotFocus
```

```
Me.TextBox5.Text = Year(Now) - Year(Me.MaskedTextBox1.Text)
```

```
End Sub
```

```
Private Sub TextBox5_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox5.TextChanged
```

```
End Sub
```

BOTON INICIO EN FORMULARIO ALUMNOS

```
Private Sub ButtonInicio_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonInicio.Click
```

```
ipos = 0
```

```
Me.muestra_datos()
```

```
mostrar_materias(odatarow("anocursa"))
```

```
End Sub
```


BOTON SIGUIENTE EN FORMULARIO ALUMNOS

```
Private Sub ButtonSiguiente_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles ButtonSiguiente.Click  
  
    If ipos = oDataSet.Tables("Alumnos").Rows.Count - 1 Then  
  
        MessageBox.Show("Ult. rec.")  
  
    Else  
  
        ipos += 1  
  
        Me.muestra_datos()  
  
        mostrar_materias(odatarow("anocursa"))  
  
    End If  
  
End Sub
```

BOTON ANTERIOR EN FORMULARIO ALUMNOS

```
Private Sub ButtonAnterior_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles ButtonAnterior.Click  
  
    If ipos = 0 Then  
  
        MessageBox.Show("Pri. rec.")  
  
    Else  
  
        ipos -= 1  
  
        Me.muestra_datos()  
  
        mostrar_materias(odatarow("anocursa"))  
  
    End If  
  
End Sub
```

BOTON ULTIMO EN FORMULARIO ALUMNOS

```
Private Sub ButtonUltimo_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonUltimo.Click

    ipos = oDataSet.Tables("Alumnos").Rows.Count - 1

    Me.muestra_datos()

    mostrar_materias(odatarow("anocursa"))

End Sub
```

BOTON DESHACER EN FORMULARIO ALUMNOS

```
Private Sub ButtonDeshacer_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonDeshacer.Click

    agregando = False

    editando = False

    Me.ButtonGuardar.Enabled = False

    Me.ButtonInicio.Enabled = True

    Me.ButtonAnterior.Enabled = True

    Me.ButtonNuevo.Enabled = True

    Me.ButtonBorrar.Enabled = True

    Me.ButtonSiguiente.Enabled = True

    Me.ButtonUltimo.Enabled = True

    Me.ButtonEditar.Enabled = True

    Me.ButtonDeshacer.Enabled = True

    muestra_datos()

    mostrar_materias(odatarow("anocursa"))

End Sub
```

```

Private Sub mostrar_materias(ByVal grad As String)

    Me.Label14.Text = ""
    Me.Label16.Text = ""
    Me.Label17.Text = ""
    Me.Label18.Text = ""
    Me.Label19.Text = ""
    Me.Label20.Text = ""
    Me.Label21.Text = ""
    Me.Label22.Text = ""
    Me.Label23.Text = ""
    Me.Label24.Text = ""

    ipos1 = 0

    oConexion = New SqlConnection("Server=(local)\SQLEXPRESS;" & _
                                   "DataBase=colegioanglicano;" & _
                                   "Integrated Security=SSPI;Connect
Timeout=5")

    Dim odataset1 As New DataSet

    oAdaptador = New SqlDataAdapter("SELECT * FROM grados where
IDGrado =' " & grad & "';", oConexion)

    oAdaptador.Fill(odataset1, "grados")

    oAdaptador = Nothing

    If odataset1.Tables("grados").Rows.Count = 0 Then
    Else

        odatarow1 = odataset1.Tables("grados").Rows(ipos1)

```

```
        XXX = odatarow1("nomenclatura")

    End If

    ipos1 = 0

    Select Case XXX

        Case "N01"

            oAdaptador = New SqlDataAdapter("SELECT * FROM materias
WHERE N01= 1;", oConexion)

        Case "N02"

            oAdaptador = New SqlDataAdapter("SELECT * FROM materias
WHERE N02=1;", oConexion)

        Case "N03"

            oAdaptador = New SqlDataAdapter("SELECT * FROM materias
WHERE N03=1;", oConexion)

        Case "N04"

            oAdaptador = New SqlDataAdapter("SELECT * FROM materias
WHERE N04=1;", oConexion)

        Case "N05"

            oAdaptador = New SqlDataAdapter("SELECT * FROM materias
WHERE N05=1;", oConexion)

        Case "N06"

            oAdaptador = New SqlDataAdapter("SELECT * FROM materias
WHERE N06=1;", oConexion)

        Case "N07"
```

```
oAdaptador = New SqlDataAdapter("SELECT * FROM materias  
WHERE N07=1;", oConexion)
```

```
Case "N08"
```

```
oAdaptador = New SqlDataAdapter("SELECT * FROM materias  
WHERE N08=1;", oConexion)
```

```
Case "N09"
```

```
oAdaptador = New SqlDataAdapter("SELECT * FROM materias  
WHERE N09=1;", oConexion)
```

```
Case "N10"
```

```
oAdaptador = New SqlDataAdapter("SELECT * FROM materias  
WHERE N10=1;", oConexion)
```

```
Case "N11"
```

```
oAdaptador = New SqlDataAdapter("SELECT * FROM materias  
WHERE N11=1;", oConexion)
```

```
Case "N12"
```

```
oAdaptador = New SqlDataAdapter("SELECT * FROM materias  
WHERE N12=1;", oConexion)
```

```
Case "N13"
```

```
oAdaptador = New SqlDataAdapter("SELECT * FROM materias  
WHERE N13=1;", oConexion)
```

```
Case "N14"
```

```
oAdaptador = New SqlDataAdapter("SELECT * FROM materias  
WHERE N14=1;", oConexion)
```

```
End Select

oAdaptador.Fill(odataset1, "materias")

oAdaptador = Nothing

If odataset1.Tables("materias").Rows.Count = 0 Then
Else
    odatarow1 = odataset1.Tables("materias").Rows(ipos1)
    For Each row As DataRow In
        odataset1.Tables("materias").Rows

        odatarow1 = odataset1.Tables("materias").Rows(ipos1)

Select Case ipos1
    Case 0
        Me.Label14.Text = odatarow1("nombremateria")
    Case 1
        Me.Label16.Text = odatarow1("nombremateria")
    Case 2
        Me.Label17.Text = odatarow1("nombremateria")
    Case 3
        Me.Label18.Text = odatarow1("nombremateria")
    Case 4
        Me.Label19.Text = odatarow1("nombremateria")
    Case 5
        Me.Label20.Text = odatarow1("nombremateria")
    Case 6
```

```

        Me.Label21.Text = odatarow1("nombremateria")
    Case 7
        Me.Label22.Text = odatarow1("nombremateria")
    Case 8
        Me.Label23.Text = odatarow1("nombremateria")
    Case 9
        Me.Label24.Text = odatarow1("nombremateria")

End Select

    ipos1 += 1

Next

End If

End Sub

```

BOTON EDITAR EN FORMULARIO ALUMNOS

```

Private Sub ButtonEditar_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ButtonEditar.Click

    Me.TextBox2.ReadOnly = False

    Me.TextBox3.ReadOnly = False

    Me.TextBox6.ReadOnly = False

    Me.MaskedTextBox1.ReadOnly = False

```

```
Me.TextBox5.ReadOnly = False
Me.TextBox8.ReadOnly = False
Me.TextBox7.ReadOnly = False
Me.TextBox10.ReadOnly = False
Me.TextBox4.ReadOnly = False
Me.CheckBox1.Enabled = True
Me.AxTDBCombo1.Enabled = True
Me.AxTDBCombo2.Enabled = True
Me.AxTDBCombo3.Enabled = True
```

```
Me.ButtonGuardar.Enabled = True
Me.ButtonInicio.Enabled = False
Me.ButtonAnterior.Enabled = False
Me.ButtonNuevo.Enabled = False
Me.ButtonBorrar.Enabled = False
Me.ButtonSiguiete.Enabled = False
Me.ButtonUltimo.Enabled = False
Me.ButtonEditar.Enabled = False
Me.ButtonDeshacer.Enabled = True
```

```
editando = True
```

```
Me.TextBox2.Focus()
```

```
End Sub
```

```
Private Sub TextBox7_GotFocus(ByVal sender As Object, ByVal e As
System.EventArgs) Handles TextBox7.GotFocus
```

```
    If Me.TextBox8.Text = "M" Or Me.TextBox8.Text = "F" Then
```

```
        Else
```



```

        Me.TextBox8.Text = ""

        Me.TextBox8.Focus()

        MessageBox.Show("Error en dato")

    End If

End Sub

Private Sub TextBox7_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TextBox7.TextChanged

End Sub

Private Sub TextBox2_GotFocus(ByVal sender As Object, ByVal e As
System.EventArgs) Handles TextBox2.GotFocus

    If agregando Then

        If Len(Trim(Me.TextBox1.Text)) < 4 Then

            Me.TextBox1.Text = ""

            Me.TextBox1.Focus()

            MessageBox.Show("Codigo no valido", "",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

            Exit Sub

        End If

        oConexion = New SqlConnection("Server=(local)\SQLEXPRESS;"
& _

        "DataBase=colegioanglicano;" & _

        "Integrated Security=SSPI;Connect Timeout=5")

        oComando = New SqlCommand

        oComando.Connection = oConexion

```

```

        oComando.CommandText = "SELECT * FROM alumnos where
idALUMNO = '" & Me.TextBox1.Text & "';"

        oAdaptador = New SqlDataAdapter(oComando)

        oCmdBuilder = New SqlCommandBuilder(oAdaptador)

        oDataSet = New DataSet

        oConexion.Open()

        oAdaptador.Fill(oDataSet, "alumnos")

        If oDataSet.Tables("alumnos").Rows.Count > 0 Then

            Me.TextBox1.Focus()

            MessageBox.Show("Codigo ya existe", "*Advertencia*",
MessageBoxButtons.OK, MessageBoxIcon.Warning)

        End If

        oConexion = New SqlConnection("Server=(local)\SQLEXPRESS;"
& _

        "DataBase=colegioanglicano;" & _

        "Integrated Security=SSPI;Connect Timeout=5")

        oComando = New SqlCommand

        oComando.Connection = oConexion

        oComando.CommandText = "SELECT * FROM alumnos;"

        oAdaptador = New SqlDataAdapter(oComando)

        oCmdBuilder = New SqlCommandBuilder(oAdaptador)

        oDataSet = New DataSet

        oConexion.Open()

        oAdaptador.Fill(oDataSet, "alumnos")

        End If

    End Sub

```

```
Private Sub TextBox2_TextChanged(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles TextBox2.TextChanged
```

```
End Sub
```

```
Private Sub TextBox1_GotFocus(ByVal sender As Object, ByVal e As  
System.EventArgs) Handles TextBox1.GotFocus
```

```
Me.TextBox1.Text = ""
```

```
y = 0
```

```
xcad = ""
```

```
num = ""
```

```
num = Str(oDataSet.Tables("Alumnos").Rows.Count + 1)
```

```
y = Len(num)
```

```
For i = y + 1 To 5
```

```
    xcad = xcad + "0"
```

```
Next
```

```
num = xcad + Trim(num)
```

```
Me.TextBox1.Text = num
```

```
End Sub
```

```
Private Sub TextBox1_TextChanged(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles TextBox1.TextChanged
```

```
End Sub
```

```
Private Sub Adodc2_Load(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Adodc2.Load
```

```

End Sub

Private Sub generar_notas()

    ipos1 = 0

    oConexion = New SqlConnection("Server=(local)\SQLEXPRESS;" & _
                                   "DataBase=colegioanglicano;" & _
                                   "Integrated Security=SSPI;Connect
Timeout=5")

    Dim odataset1 As New DataSet

    oAdaptador = New SqlDataAdapter("SELECT * FROM grados where
IDGrado =' & Me.AxTDBCombo3.BoundText & '";", oConexion)

    oAdaptador.Fill(odataset1, "grados")

    oAdaptador = Nothing

    If odataset1.Tables("grados").Rows.Count = 0 Then
    Else

        odatarow1 = odataset1.Tables("grados").Rows(ipos1)

        XXX = odatarow1("nomenclatura")

    End If

    ipos1 = 0

    Select Case XXX

        Case "N01"

            oAdaptador = New SqlDataAdapter("SELECT * FROM materias
WHERE N01= 1;", oConexion)

        Case "N02"

            oAdaptador = New SqlDataAdapter("SELECT * FROM materias
WHERE N02=1;", oConexion)

        Case "N03"

```

```
oAdaptador = New SqlDataAdapter("SELECT * FROM materias  
WHERE N03=1;", oConexion)
```

```
Case "N04"
```

```
oAdaptador = New SqlDataAdapter("SELECT * FROM materias  
WHERE N04=1;", oConexion)
```

```
Case "N05"
```

```
oAdaptador = New SqlDataAdapter("SELECT * FROM materias  
WHERE N05=1;", oConexion)
```

```
Case "N06"
```

```
oAdaptador = New SqlDataAdapter("SELECT * FROM materias  
WHERE N06=1;", oConexion)
```

```
Case "N07"
```

```
oAdaptador = New SqlDataAdapter("SELECT * FROM materias  
WHERE N07=1;", oConexion)
```

```
Case "N08"
```

```
oAdaptador = New SqlDataAdapter("SELECT * FROM materias  
WHERE N08=1;", oConexion)
```

```
Case "N09"
```

```
oAdaptador = New SqlDataAdapter("SELECT * FROM materias  
WHERE N09=1;", oConexion)
```

```
Case "N10"
```

```
oAdaptador = New SqlDataAdapter("SELECT * FROM materias  
WHERE N10=1;", oConexion)
```

```
Case "N11"
```

```
oAdaptador = New SqlDataAdapter("SELECT * FROM materias  
WHERE N11=1;", oConexion)
```

```
Case "N12"
```

```
oAdaptador = New SqlDataAdapter("SELECT * FROM materias  
WHERE N12=1;", oConexion)
```

```
Case "N13"
```

```
oAdaptador = New SqlDataAdapter("SELECT * FROM materias  
WHERE N13=1;", oConexion)
```

```

        Case "N14"

            oAdaptador = New SqlDataAdapter("SELECT * FROM materias
WHERE N14=1;", oConexion)

        End Select

oAdaptador.Fill(odataset1, "materias")

oAdaptador = Nothing

If odataset1.Tables("materias").Rows.Count = 0 Then

Else

    odatarow1 = odataset1.Tables("materias").Rows(ipos1)

    Dim cmat As String, nmat As String

    Dim ssql As String

    Dim objcmd As New SqlCommand

    oConexion.Open()

    For Each row As DataRow In
odataset1.Tables("materias").Rows

        odatarow1 = odataset1.Tables("materias").Rows(ipos1)

        cmat = odatarow1("idmateria")

        nmat = odatarow1("nombremateria")

        ssql = "INSERT INTO NOTAS
(IDALUMNO,IDASIGNATURA,nombremateria) VALUES ('" & Me.TextBox1.Text &
"' , '" & cmat & "' , '" & nmat & "');"

        objcmd = New SqlCommand(ssql, oConexion)

```

```
        objcmd.ExecuteNonQuery()  
        ipos1 += 1  
  
    Next  
  
End If  
  
End Sub
```

CODIGO PARA CARGAR EL FORMULARIO ASIGNAR GRUPO EN EL MENU PRINCIPAL

```
Private Sub AsignarGrupoToolStripMenuItem_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
AsignarGrupoToolStripMenuItem.Click  
  
    frmAsignarGrupo.ShowDialog()  
  
    frmAsignarGrupo.Close()  
  
End Sub
```

The screenshot shows a dialog box titled "frmAsignarGrupo". It features the following elements:

- A dropdown menu labeled "Nombres y Apellidos" with the control name "AxTDBCombo1".
- A set of navigation buttons labeled "Adodc1" with left and right arrow icons.
- A text field labeled "Grado" followed by a dotted line.
- A text field labeled "Año Lectivo" followed by a dotted line.
- A dropdown menu labeled "Grupo" with the control name "AxTDBCombo2".
- A "Guardar" button with a floppy disk icon.
- A second set of navigation buttons labeled "Adodc2" with left and right arrow icons.

FORMULARIO ASIGNAR GRUPO

```
Imports System.Data
```

```
Imports System.Data.SqlClient
```

```
Public Class frmAsignarGrupo
```

```
    Dim odataRow As DataRow
```

```
    Private ipos As Integer = 0
```

```
    Dim oConexion As New SqlConnection
```

```
    Dim oDataSet As DataSet
```

```
    Dim oAdaptador As SqlDataAdapter
```

```
    Dim igrado As String
```

```
    Private Sub AxTDBCombol_KeyDownEvent(ByVal sender As Object, ByVal  
e As AxTrueOleDbList80.TrueDBComboEvents_KeyDownEvent) Handles  
AxTDBCombol_KeyDownEvent
```

```
        If e.keyCode = 13 Then
```

```
            oConexion =NewSqlConnection("Server=(local)\SQLEXPRESS;" &  
"DataBase=colegioanglicano;" &  
"Integrated Security=SSPI;Connect Timeout=5")
```

```
            oAdaptador = New SqlDataAdapter("SELECT * FROM alumnos where idalumno  
= '" & Me.AxTDBCombol.BoundText & "';", oConexion)
```

```
                oDataSet = New DataSet
```

```
                oAdaptador.Fill(oDataSet, "Alumnos")
```

```
                oAdaptador = Nothing
```



```

        If oDataSet.Tables("alumnos").Rows.Count = 0 Then
            MessageBox.Show("No hay matriculas registradas", "",
                MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
        Else
            odatarow = oDataSet.Tables("alumnos").Rows(i)
            igrado = odatarow("anocursa")
            Me.Label4.Text = odatarow("anolectivo")

        End If

        oAdaptador = New SqlDataAdapter("SELECT * FROM grados where
            idgrado = '" & igrado & "'", oConexion)
        oAdaptador.Fill(oDataSet, "grados")
        oAdaptador = Nothing

        If oDataSet.Tables("grados").Rows.Count = 0 Then
            MessageBox.Show("Error, idgrado no existe en tabla
                grados", "", MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
        Else
            odatarow = oDataSet.Tables("grados").Rows(i)
            Me.Label5.Text = odatarow("nombregrado")

        End If

        Me.AxTDBCombo2.Focus()

    End If
End Sub

```

```
Private Sub frmAsignarGrupo_Activated(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Activated
```

```
Me.AxTDBCombo1.Focus()
```

```
End Sub
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
```

```
oConexion.Open()
```

```
Dim ssql As String = "Update Alumnos set idgrupo = '" &  
Me.AxTDBCombo2.BoundText & "' where idalumno = '" &  
Me.AxTDBCombo1.BoundText & "';"
```

```
Dim objcmd As New SqlCommand(ssql, oConexion)
```

```
objcmd.ExecuteNonQuery()
```

```
MessageBox.Show("Datos actualizados")
```

```
End Sub
```

```
Private Sub frmAsignarGrupo_Deactivate(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Deactivate
```

```
oConexion.Close()
```

```
End Sub
```

```
Private Sub frmAsignarGrupo_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
Me.AxTDBCombo1.Enabled = True
```

```
Me.AxTDBCombo2.Enabled = True
```

```
End Sub
```

```
End Class
```

EN LA BARRA DE HERRAMIENTAS DEL MENU PRINCIPAL EN NOTAS
CODIGO PARA CARGAR EL FORMULARIO ACTUALIZAR NOTAS EN EL MENU
PRINCIPAL

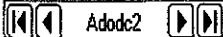
```
Private Sub ActualizarNotasToolStripMenuItem_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
ActualizarNotasToolStripMenuItem.Click
```

```
    frmNotas.ShowDialog()
```

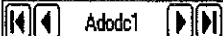
```
    frmNotas.Close()
```

```
End Sub
```

Actualizar Notas

Seleccione el Nombre del Estudiante  Adodc2

IdAlumno	Materia	1erParcial IS	2doParcial IS	3erParcial IS	FINAL IS	1erParcial IIS	2doParcial IIS	3erParcial IIS
*								

 Adodc1

FORMULARIO NOTAS

```
Public Class frmNotas
```

```
    Private Sub frmNotas_Load(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles MyBase.Load
```

```
        Me.AxTDBGrid1.Columns(0).Caption = "Id Alumno"
```

```
        Me.AxTDBGrid1.Columns(0).Width = 50
```

```
        Me.AxTDBGrid1.Columns(2).Width = 80
```

```
        Me.AxTDBGrid1.Columns(3).Width = 80
```

```
        Me.AxTDBGrid1.Columns(4).Width = 80
```

```
        Me.AxTDBGrid1.Columns(5).Width = 80
```

```
        Me.AxTDBGrid1.Columns(6).Width = 80
```

```
        Me.AxTDBGrid1.Columns(7).Width = 80
```

```
        Me.AxTDBGrid1.Columns(8).Width = 80
```

```
        Me.AxTDBGrid1.Columns(9).Width = 80
```

```
        Me.AxTDBGrid1.Columns(10).Width = 80
```

```
        Me.AxTDBGrid1.Columns(2).NumberFormat = "##0"
```

```
        Me.AxTDBGrid1.Columns(3).NumberFormat = "##0"
```

```
        Me.AxTDBGrid1.Columns(4).NumberFormat = "##0"
```

```
        Me.AxTDBGrid1.Columns(5).NumberFormat = "##0"
```

```
        Me.AxTDBGrid1.Columns(6).NumberFormat = "##0"
```

```
        Me.AxTDBGrid1.Columns(7).NumberFormat = "##0"
```

```
        Me.AxTDBGrid1.Columns(8).NumberFormat = "##0"
```

```
        Me.AxTDBGrid1.Columns(9).NumberFormat = "##0"
```

```
        Me.AxTDBGrid1.Columns(10).NumberFormat = "##0"
```

```
        Me.AxTDBGrid1.Columns(5).Locked = True
```

```
Me.AxTDBGrid1.Columns(9).Locked = True  
Me.AxTDBGrid1.Columns(10).Locked = True
```

```
End Sub
```

```
Private Sub AxTDBGrid1_AfterColUpdate(ByVal sender As  
System.Object, ByVal e As  
AxTrueOleDBGrid80.TrueDBGridEvents_AfterColUpdateEvent) Handles  
AxTDBGrid1.AfterColUpdate
```

```
End Sub
```

```
Private Sub AxTDBGrid1_BeforeRowColChange(ByVal sender As Object,  
ByVal e As AxTrueOleDBGrid80.TrueDBGridEvents_BeforeRowColChangeEvent)  
Handles AxTDBGrid1.BeforeRowColChange
```

```
Dim a As Double, b As Double, c As Double
```

```
Select Case Me.AxTDBGrid1.Col
```

```
Case 2
```

```
If Me.AxTDBGrid1.Columns(2).Value < 0 Or  
Me.AxTDBGrid1.Columns(2).Value > 100 Then
```

```
MessageBox.Show("Valor no valido")
```

```
e.cancel = True
```

```
End If
```

```
a = Me.AxTDBGrid1.Columns(2).Value
```

```
b = Me.AxTDBGrid1.Columns(3).Value
```

```
c = Me.AxTDBGrid1.Columns(4).Value
```

```
Me.AxTDBGrid1.Columns(5).Value = (a + b + c) / 3
```

```
Case 3
```

```
If Me.AxTDBGrid1.Columns(3).Value < 0 Or  
Me.AxTDBGrid1.Columns(3).Value > 100 Then
```

```
    MessageBox.Show("Valor no valido")
```

```
    e.cancel = True
```

```
End If
```

```
a = Me.AxTDBGrid1.Columns(2).Value
```

```
b = Me.AxTDBGrid1.Columns(3).Value
```

```
c = Me.AxTDBGrid1.Columns(4).Value
```

```
Me.AxTDBGrid1.Columns(5).Value = (a + b + c) / 3
```

```
Case 4
```

```
If Me.AxTDBGrid1.Columns(4).Value < 0 Or  
Me.AxTDBGrid1.Columns(4).Value > 100 Then
```

```
    MessageBox.Show("Valor no valido")
```

```
    e.cancel = True
```

```
End If
```

```
a = Me.AxTDBGrid1.Columns(2).Value
```

```
b = Me.AxTDBGrid1.Columns(3).Value
```

```
c = Me.AxTDBGrid1.Columns(4).Value
```

```
Me.AxTDBGrid1.Columns(5).Value = (a + b + c) / 3
```

```
Case 6
```

```
If Me.AxTDBGrid1.Columns(6).Value < 0 Or  
Me.AxTDBGrid1.Columns(6).Value > 100 Then
```

```
    MessageBox.Show("Valor no valido")
```

```
e.cancel = True
```

End If

a = Me.AxTDBGrid1.Columns(6).Value

b = Me.AxTDBGrid1.Columns(7).Value

c = Me.AxTDBGrid1.Columns(8).Value

Me.AxTDBGrid1.Columns(9).Value = (a + b + c) / 3

Case 7

If Me.AxTDBGrid1.Columns(7).Value < 0 Or
Me.AxTDBGrid1.Columns(7).Value > 100 Then

MessageBox.Show("Valor no valido")

e.cancel = True

End If

a = Me.AxTDBGrid1.Columns(6).Value

b = Me.AxTDBGrid1.Columns(7).Value

c = Me.AxTDBGrid1.Columns(8).Value

Me.AxTDBGrid1.Columns(9).Value = (a + b + c) / 3

Case 8

If Me.AxTDBGrid1.Columns(8).Value < 0 Or
Me.AxTDBGrid1.Columns(8).Value > 100 Then

MessageBox.Show("Valor no valido")

e.cancel = True

End If

a = Me.AxTDBGrid1.Columns(6).Value

b = Me.AxTDBGrid1.Columns(7).Value

c = Me.AxTDBGrid1.Columns(8).Value

Me.AxTDBGrid1.Columns(9).Value = (a + b + c) / 3

```

End Select

a = Me.AxTDBGrid1.Columns(5).Value

b = Me.AxTDBGrid1.Columns(9).Value

Me.AxTDBGrid1.Columns(10).Value = (a + b) / 2

End Sub

Private Sub AxTDBCombol_Enter(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles AxTDBCombol.Enter

End Sub

Private Sub AxTDBCombol_KeyDownEvent(ByVal sender As Object, ByVal
e As AxTrueOleDbList80.TrueDBComboEvents_KeyDownEvent) Handles
AxTDBCombol.KeyDownEvent

If e.keyCode = 13 Then

    Me.Adodc1.RecordSource = "Select * from vista_notas where
idalumno = '" & Me.AxTDBCombol.BoundText & "';"

    Me.Adodc1.Refresh()

    Me.AxTDBGrid1.DataSource = Me.Adodc1

    Me.AxTDBGrid1.Refresh()

    Me.AxTDBGrid1.Visible = True

End If

End Sub

Private Sub Label1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label1.Click

End Sub

End Class

```


EN LA BARRA DE HERRAMIENTAS DEL MENU PRINCIPAL EN REPORTES

CODIGO PARA CARGAR EL FORMULARIO BOLETINES EN EL MENU PRINCIPAL

```
Private Sub BoletinesToolStripMenuItem_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
BoletinesToolStripMenuItem.Click
```

```
    FRMImpNotas.ShowDialog()
```

```
    FRMImpNotas.Close()
```

```
End Sub
```

Current Page No.: 1 Total Page No.: 1+ Zoom Factor: 100%

CODIGO DE VISTA BOLETIN EN MENU PRINCIPAL

```
Imports System.Data

Imports System.Data.SqlClient

Public Class VistaBoletin

    Private Sub VistaBoletin_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load

        Dim rpt As New boletin()

        Dim myConnection As New SqlConnection()

        Dim MyCommand As New SqlCommand()

        Dim myDA As New SqlDataAdapter()

        Dim myDS As New ColegioDataSet()

        myConnection = New
SqlConnection("Server=(local)\SQLEXPRESS;DataBase=colegioanglicano;Inte
grated Security=SSPI;Connect Timeout=5")

        '*****
        *

        'UTILIZAMOS ESTE CODIGO PARA LIMPIAR LA TABLA BOLETIN, LA CUAL
ES UNA TABLA TEMPORAL

        'QUE SOLO SE USA PARA IMPRIMIR LAS NOTAS. EN LA PRIMERA PARTE
LIMPIAMOS LA TABLA CON LA

        'INSTRUCCION DELETE, LUEGO LA VOLVEMOS A LLENAR CON LA INS.
INSERT INTO DESDE LA VISTA

        'VISTA_BOLETIN

        Dim ssql As String

        Dim objcmd As New SqlCommand
```

```
myConnection.Open()

ssql = "delete from boletin" 'limpiamos la tabla

objcmd = New SqlCommand(ssql, myConnection)

objcmd.ExecuteNonQuery()

myConnection.Close()
```

```
myConnection.Open()

ssql = "INSERT INTO BOLETIN
(IDALUMNO, nombres, APELLIDOS, nombremateria, nombregrado, nombregrupo, nombre
turno, PrimerParcialISemestre, " & _

        "SegundoParcialISemestre, TercerParcialISemestre,
NotaFinalISemestre, PrimerParcialIISemestre, SegundoParcialIISemestre,
" & _

        "TercerParcialIISemestre, NotaFinalIISemestre,
NotaFinal) " & _

        "SELECT
IDALUMNO, NOMBRES, APELLIDOS, NOMBREMATERIA, nombregrado, nombregrupo, nombre
turno, PrimerParcialISemestre, " & _

        "SegundoParcialISemestre, TercerParcialISemestre,
NotaFinalISemestre, PrimerParcialIISemestre, SegundoParcialIISemestre,"
& _

        "TercerParcialIISemestre, NotaFinalIISemestre, NotaFinal
FROM VISTA_BOLETIN where anocursa = '" &
FRMImpNotas.AxTDBCombol.BoundsText & "';"

objcmd = New SqlCommand(ssql, myConnection)

objcmd.ExecuteNonQuery()

myConnection.Close()
```

```
! *****
**
```

```
MyCommand.Connection = myConnection
```

```
MyCommand.CommandText = "SELECT * FROM Boletin"

MyCommand.CommandType = CommandType.Text

myDA.SelectCommand = MyCommand

myDA.Fill(myDS, "Boletin")

rpt.SetDataSource(myDS)

CrystalReportViewer1.ReportSource = rpt
```

```
End Sub
```

```
Private Sub CrystalReportViewer1_Load(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
CrystalReportViewer1.Load
```

```
End Sub
```

```
End Class
```

CODIGO PARA CARGAR LA VISTA PLANILLA EN EL MENU PRINCIPAL

```
Private Sub PlanillaToolStripMenuItem_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
PlanillaToolStripMenuItem.Click
```

```
    vistaplantilla.ShowDialog()
```

```
End Sub
```

19/11/2007

COLEGIO ANGLICANO
PUERTO CABEZAS - RAAN
PLANTILLA DE ESTUDIANTES

Saturday
Drill Down

Time		XY Scatter
Total Grupo	1	

OLAP

Fiel	Boolean	XY Scatter
Total Grupo	1	

Total Grado 2

Thursday
Standard

DateTime		Stock
Total Grupo	1	

Total Grado 1

Current Page No.: 1 Total Page No.: 1+ Zoom Factor: 100%

CODIGO VISTA PLANILLA EN MENU PRINCIPAL

```
Imports System.Data
```

```
Imports System.Data.SqlClient
```

```
Public Class vistaplantilla
```

```
    Private Sub vistaplantilla_Load(ByVal sender As Object, ByVal e As  
System.EventArgs) Handles Me.Load
```

```
        Dim rpt As New Plantilla()
```

```
        Dim myConnection As New SqlConnection()
```

```
        Dim MyCommand As New SqlCommand()
```

```
        Dim myDA As New SqlDataAdapter()
```

```
        Dim myDS As New ColegioDataSet()
```

```
        myConnection = New  
SqlConnection("Server=(local)\SQLEXPRESS;DataBase=colegioanglicano;Inte  
grated Security=SSPI;Connect Timeout=5")
```

```
        MyCommand.Connection = myConnection
```

```
        MyCommand.CommandText = "SELECT * FROM Vista_Plantilla"
```

```
        MyCommand.CommandType = CommandType.Text
```

```
        myDA.SelectCommand Imports System.Data
```

```
Imports System.Data.SqlClient
```

```
Public Class vistaplantilla
```

```
    Private Sub vistaplantilla_Load(ByVal sender As Object, ByVal e As  
System.EventArgs) Handles Me.Load
```

```

Dim rpt As New Plantilla()

Dim myConnection As New SqlConnection()

Dim MyCommand As New SqlCommand()

Dim myDA As New SqlDataAdapter()

Dim myDS As New ColegioDataSet()

myConnection = New
SqlConnection("Server=(local)\SQLEXPRESS;DataBase=colegioanglicano;Inte
grated Security=SSPI;Connect Timeout=5")

MyCommand.Connection = myConnection

MyCommand.CommandText = "SELECT * FROM Vista_Plantilla"

MyCommand.CommandType = CommandType.Text

myDA.SelectCommand = MyCommand

myDA.Fill(myDS, "Vista_Plantilla")

rpt.SetDataSource(myDS)

CrystalReportViewer1.ReportSource = rpt

End Sub

End Classommand = MyCommand

myDA.Fill(myDS, "Vista_Plantilla")

rpt.SetDataSource(myDS)

CrystalReportViewer1.ReportSource = rpt

End Sub

End Class

```

EN LA BARRA DE HERRAMIENTAS DEL MENU PRINCIPAL EN CATALOGOS

CODIGO PARA CARGAR EL FORMULARIO MATERIAS EN EL MENU PRINCIPAL

```
Private Sub MateriasToolStripMenuItem_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
MateriasToolStripMenuItem.Click
```

```
    frmMaterias.ShowDialog()
```

```
End Sub
```

Catálogo de Materias

Id Materia

Nombre materia

<input type="checkbox"/> 1er.Nivel	<input type="checkbox"/> 5to.Grado
<input type="checkbox"/> 2do.Nivel	<input type="checkbox"/> 6to.Grado
<input type="checkbox"/> 3er.Nivel	<input type="checkbox"/> 1er.Año
<input type="checkbox"/> 1er.Grado	<input type="checkbox"/> 2do.Año
<input type="checkbox"/> 2do.Grado	<input type="checkbox"/> 3er.Año
<input type="checkbox"/> 3er.Grado	<input type="checkbox"/> 4to.Año
<input type="checkbox"/> 4to.Grado	<input type="checkbox"/> 5to.Año

Barra de Botones

Label3

CODIGO FORMULARIO MATERIAS

```
Imports System.Data
```

```
Imports System.Data.SqlClient
```

```
Public Class frmMaterias
```

```
    Dim oDataRow As DataRow
```

```
    Private ipos As Integer = 0
```

```
    Dim oDataSet As DataSet
```

```
    Dim oAdaptador As SqlDataAdapter
```

```
    Dim agregando As Boolean, editando As Boolean
```

```
    Dim oCmdBuilder As SqlCommandBuilder
```

```
    Dim oComando As SqlCommand
```

```
    Dim oConexion As New SqlConnection
```

```
    Private Sub frmMaterias_FormClosing(ByVal sender As Object, ByVal e  
As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
```

```
        If agregando Then
```

```
            Dim Resp As MsgBoxResult
```

```
            resp = MsgBox("Desea salir", MsgBoxStyle.YesNo)
```

```
            If Resp = MsgBoxResult.Yes Then
```

```
                agregando = False
```

```
                e.Cancel = False
```

```
                Me.Close()
```

```

        Else
            e.Cancel = True
        End If
    End If

    If editando Then

        Dim Resp As MsgBoxResult

        Resp = MsgBox("Desea salir", MsgBoxStyle.YesNo)

        If Resp = MsgBoxResult.Yes Then

            editando = False

            Me.Close()

            e.Cancel = False

        Else

            e.Cancel = True

        End If

    End If

End Sub

Private Sub frmMaterias_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load

    'este codigo es para actualizar el grid

    oConexion = New SqlConnection("Server=(local)\SQLEXPRESS;" & _
    "DataBase=colegioanglicano;" & _
    "Integrated Security=SSPI;Connect Timeout=5")

    oComando = New SqlCommand

    oComando.Connection = oConexion

```

```

oComando.CommandText = "SELECT * FROM Materias"
oAdaptador = New SqlDataAdapter(oComando)
oCmdBuilder = New SqlCommandBuilder(oAdaptador)
oDataSet = New DataSet
oConexion.Open()
oAdaptador.Fill(oDataSet, "Materias")
If oDataSet.Tables("materias").Rows.Count = 0 Then
Else

    muestra_datos()

    Me.Label3.Text = "Recs. " & Me.ipos + 1 & " de " &
oDataSet.Tables("materias").Rows.Count
End If

Me.ButtonGuardar.Enabled = False
Me.ButtonInicio.Enabled = True
Me.ButtonAnterior.Enabled = True
Me.ButtonNuevo.Enabled = True
Me.ButtonBorrar.Enabled = True
Me.ButtonSiguiente.Enabled = True
Me.ButtonUltimo.Enabled = True
Me.ButtonEditar.Enabled = True
Me.ButtonDeshacer.Enabled = True
Me.TextBox1.ReadOnly = True
Me.TextBox2.ReadOnly = True
Me.TextBox1.Focus()
agregando = False

```

```
        editando = False

        End Sub

        Public Sub muestra_datos()

odatarow = oDataSet.Tables("materias").Rows(ipos)

Me.TextBox1.Text = odatarow("Idmateria")

Me.TextBox2.Text = odatarow("Nombremateria")

If odatarow("n01") = True Then

        Me.CheckBox1.Checked = True

Else

        Me.CheckBox1.Checked = False

End If

If odatarow("n02") = True Then

        Me.CheckBox2.Checked = True

Else

        Me.CheckBox2.Checked = False

End If

If odatarow("n03") = True Then

        Me.CheckBox3.Checked = True

Else

        Me.CheckBox3.Checked = False

End If

If odatarow("n04") = True Then

        Me.CheckBox4.Checked = True

Else
```

```
        Me.CheckBox4.Checked = False
    End If
```

```
If odatarow("n05") = True Then
    Me.CheckBox5.Checked = True
Else
    Me.CheckBox5.Checked = False
End If
```

```
If odatarow("n06") = True Then
    Me.CheckBox6.Checked = True
Else
    Me.CheckBox6.Checked = False
End If
```

```
If odatarow("n07") = True Then
    Me.CheckBox7.Checked = True
Else
    Me.CheckBox7.Checked = False
End If
```

```
If odatarow("n08") = True Then
    Me.CheckBox8.Checked = True
Else
    Me.CheckBox8.Checked = False
End If
```

```
If odatarow("n09") = True Then
    Me.CheckBox9.Checked = True
Else
    Me.CheckBox9.Checked = False
End If
```

```
If odatarow("n10") = True Then
    Me.CheckBox10.Checked = True

Else
    Me.CheckBox10.Checked = False
End If
```

```
If odatarow("n11") = True Then
    Me.CheckBox11.Checked = True
Else
    Me.CheckBox11.Checked = False
End If
```

```
If odatarow("n12") = True Then
    Me.CheckBox12.Checked = True
Else
    Me.CheckBox12.Checked = False
End If
```

```
If odatarow("n13") = True Then
    Me.CheckBox13.Checked = True
```

```
Else
    Me.CheckBox13.Checked = False
End If

If odatarow("n14") = True Then
    Me.CheckBox14.Checked = True
Else
    Me.CheckBox14.Checked = False
End If
```

BOTON INICIO

```
End Sub

Private Sub ButtonInicio_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonInicio.Click
    ipos = 0
    Me.muestra_datos()
    Me.Label3.Text = "Recs. " & Me.ipos + 1 & " de " & oDataSet.Tables("materias").Rows.Count
End Sub
```

BOTON SIGUIENTE

```
Private Sub ButtonSiguiente_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonSiguiente.Click
    If ipos = oDataSet.Tables("materias").Rows.Count - 1 Then
        MessageBox.Show("Ult. rec.")
    End If
End Sub
```

```

Else
    ipos += 1
    Me.muestra_datos()
End If

Me.Label3.Text = "Recs. " & Me.ipos + 1 & " de " &
oDataSet.Tables("materias").Rows.Count

End Sub

```

BOTON ANTERIOR

```

Private Sub ButtonAnterior_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ButtonAnterior.Click

    If ipos = 0 Then

        MessageBox.Show("Pri. rec.")

    Else

        ipos -= 1

        Me.muestra_datos()

    End If

    Me.Label3.Text = "Recs. " & Me.ipos + 1 & " de " &
oDataSet.Tables("materias").Rows.Count

End Sub

```

BOTON GUARDAR

```

Private Sub ButtonGuardar_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ButtonGuardar.Click

    Dim nuevo As New classMaterias

```



```

If agregando Then
    ' se instancia la variable nuevo al sub agregar
    nuevo.agregar_Materia(Me.TextBox1.Text, Me.TextBox2.Text)
End If

If editando Then
    Dim editar As New classMaterias
    editar.MODIFICAR_MATERIA(Me.TextBox1.Text,
Me.TextBox2.Text, ipos)
End If

refrescar()

Me.ButtonGuardar.Enabled = False
Me.ButtonInicio.Enabled = True
Me.ButtonAnterior.Enabled = True
Me.ButtonNuevo.Enabled = True
Me.ButtonBorrar.Enabled = True
Me.ButtonSiguiete.Enabled = True
Me.ButtonUltimo.Enabled = True
Me.ButtonEditar.Enabled = True
Me.ButtonDeshacer.Enabled = True
Me.TextBox1.ReadOnly = True
Me.TextBox2.ReadOnly = True
agregando = False

editando = False

End Sub

```

BOTON ULTIMO

```
Private Sub ButtonUltimo_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonUltimo.Click

    ipos = oDataSet.Tables("materias").Rows.Count - 1

    Me.muestra_datos()

    Me.Label3.Text = "Recs. " & Me.ipos + 1 & " de " &
oDataSet.Tables("materias").Rows.Count

End Sub
```

BOTON NUEVO

```
Private Sub ButtonNuevo_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonNuevo.Click

    Me.TextBox1.ReadOnly = False

    Me.TextBox2.ReadOnly = False

    Me.ButtonGuardar.Enabled = True

    Me.ButtonInicio.Enabled = False

    Me.ButtonAnterior.Enabled = False

    Me.ButtonNuevo.Enabled = False

    Me.ButtonBorrar.Enabled = False

    Me.ButtonSiguiente.Enabled = False

    Me.ButtonUltimo.Enabled = False

    Me.ButtonEditar.Enabled = False

    Me.ButtonDeshacer.Enabled = True
```

```
agregando = True
Me.TextBox1.Text = ""
Me.TextBox2.Text = ""

Me.CheckBox1.Checked = False
Me.CheckBox2.Checked = False
Me.CheckBox3.Checked = False
Me.CheckBox4.Checked = False
Me.CheckBox5.Checked = False
Me.CheckBox6.Checked = False
Me.CheckBox7.Checked = False
Me.CheckBox8.Checked = False
Me.CheckBox9.Checked = False
Me.CheckBox10.Checked = False
Me.CheckBox11.Checked = False
Me.CheckBox12.Checked = False
Me.CheckBox13.Checked = False
Me.CheckBox14.Checked = False

Me.TextBox1.Focus()
```

End Sub

BOTON EDITAR

```
Private Sub ButtonEditor_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonEditor.Click
```

```
    Me.TextBox2.ReadOnly = False  
    Me.ButtonGuardar.Enabled = True  
    Me.ButtonInicio.Enabled = False  
    Me.ButtonAnterior.Enabled = False  
    Me.ButtonNuevo.Enabled = False  
    Me.ButtonBorrar.Enabled = False  
    Me.ButtonSiguiente.Enabled = False  
    Me.ButtonUltimo.Enabled = False  
    Me.ButtonEditar.Enabled = False  
    Me.ButtonDeshacer.Enabled = True
```

```
    editando = True  
    Me.TextBox2.Focus()
```

```
End Sub
```

BOTON BORRAR

```
Private Sub ButtonBorrar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonBorrar.Click
```

```
    Dim borrar As String
```

```
borrar = MsgBox("Desea eliminar el registro", MsgBoxStyle.YesNo  
+ MsgBoxStyle.Question, "Advertencia")
```

```
If borrar = MsgBoxResult.Yes Then
```

```
Dim eliminar As New classMaterias
```

```
eliminar.borrar_Materia(ipos)
```

```
refrescar()
```

```
End If
```

```
End Sub
```

```
Private Sub refrescar()
```

```
'codigo para refrescar el grid con el nuevo item agregado
```

```
oConexion = New SqlConnection("Server=(local)\SQLEXPRESS;" & _
```

```
"DataBase=colegioanglicano;" & _
```

```
"Integrated Security=SSPI;Connect Timeout=5")
```

```
oComando = New SqlCommand
```

```
oComando.Connection = oConexion
```

```
oComando.CommandText = "SELECT * FROM Materias"
```

```
oAdaptador = New SqlDataAdapter(oComando)
```

```
oCmdBuilder = New SqlCommandBuilder(oAdaptador)
```

```
oDataSet = New DataSet
```

```
oConexion.Open()
```

```
oAdaptador.Fill(oDataSet, "Materias")
```

```
ipos = 0
```

```
odatarow = oDataSet.Tables("materias").Rows(ipos)
```

```
Me.muestra_datos()
```

```
Me.Label3.Text = "Recs. " & Me.ipos + 1 & " de " &  
oDataSet.Tables("materias").Rows.Count
```

```
End Sub
```

BOTON DESHACER

```
Private Sub ButtonDeshacer_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles ButtonDeshacer.Click
```

```
    Me.ButtonGuardar.Enabled = False  
    Me.ButtonInicio.Enabled = True  
    Me.ButtonAnterior.Enabled = True  
    Me.ButtonNuevo.Enabled = True  
    Me.ButtonBorrar.Enabled = True  
    Me.ButtonSiguiete.Enabled = True  
    Me.ButtonUltimo.Enabled = True  
    Me.ButtonEditar.Enabled = True  
    Me.ButtonDeshacer.Enabled = True  
    Me.TextBox1.ReadOnly = True  
    Me.TextBox2.ReadOnly = True  
    refrescar()
```

```
End Sub
```

```
Private Sub TextBox2_GotFocus(ByVal sender As Object, ByVal e As  
System.EventArgs) Handles TextBox2.GotFocus
```

```
    If agregando Then
```

```
        oConexion = New SqlConnection("Server=(local)\SQLEXPRESS;"  
& _  
        "DataBase=colegioanglicano;" & _  
        "Integrated Security=SSPI;Connect Timeout=5")  
        oComando = New SqlCommand
```

```

oComando.Connection = oConexion

oComando.CommandText = "SELECT * FROM Materias where
idmateria = '" & Me.TextBox1.Text & "';"

oAdaptador = New SqlDataAdapter(oComando)

oCmdBuilder = New SqlCommandBuilder(oAdaptador)

oDataSet = New DataSet

oConexion.Open()

oAdaptador.Fill(oDataSet, "Materias")

If oDataSet.Tables("materias").Rows.Count > 0 Then

    Me.TextBox1.Focus()

    MessageBox.Show("Codigo ya existe", "*Advertencia*",
    MessageBoxButtons.OK, MessageBoxIcon.Warning)

End If

oConexion = New SqlConnection("Server=(local)\SQLEXPRESS;"
& _

"DataBase=colegioanglicano;" & _

"Integrated Security=SSPI;Connect Timeout=5")

oComando = New SqlCommand

oComando.Connection = oConexion

oComando.CommandText = "SELECT * FROM Materias;"

oAdaptador = New SqlDataAdapter(oComando)

oCmdBuilder = New SqlCommandBuilder(oAdaptador)

oDataSet = New DataSet

oConexion.Open()

oAdaptador.Fill(oDataSet, "Materias")

End If

End Sub

```

```
Private Sub TextBox2_TextChanged(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles TextBox2.TextChanged
```

```
End Sub
```

```
End Class
```

CODIGO PARA CARGAR EL FORMULARIO GRADOS EN EL MENU PRINCIPAL

```
Private Sub GradosToolStripMenuItem_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
GradosToolStripMenuItem.Click
```

```
frmGrados.ShowDialog()
```

```
End Sub
```

Catálogo de Grados

Id Grado

Nombre Grado

Nomenclatura

Barra de Botones

⏪ ⏩ 📄 🔄 ✕ ⏪ ⏩ ⏪ ⏩ 🔍

Vista Previa

CODIGO FORMULARIO GRADOS

```
Imports System.Data

Imports System.Data.SqlClient

Public Class frmGrados

    Dim oDataRow As DataRow

    Private ipos As Integer = 0

    Dim oDataSet As DataSet

    Dim oAdaptador As SqlDataAdapter

    Dim agregando As Boolean, editando As Boolean

    Dim oCmdBuilder As SqlCommandBuilder

    Dim oComando As SqlCommand

    Dim oConexion As New SqlConnection

    Private Sub frmGrados_FormClosing(ByVal sender As Object, ByVal e
As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

        If agregando Then

            Dim Resp As MsgBoxResult

            Resp = MsgBox("Desea salir", MsgBoxStyle.YesNo)

            If Resp = MsgBoxResult.Yes Then

                agregando = False

                e.Cancel = False

                Me.Close()

            Else

                e.Cancel = True

            End If

        End If

    End Sub

End Class
```

```

        End If

End If

If editando Then

    Dim Resp As MsgBoxResult

    Resp = MsgBox("Desea salir", MsgBoxStyle.YesNo)

    If Resp = MsgBoxResult.Yes Then

        editando = False

        Me.Close()

        e.Cancel = False

    Else

        e.Cancel = True

    End If

End If

End Sub

Private Sub frmGrados_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load

    'este codigo es para actualizar el grid

    oConexion = New SqlConnection("Server=(local)\SQLEXPRESS;" & _
    "DataBase=colegioanglicano;" & _
    "Integrated Security=SSPI;Connect Timeout=5")

    oComando = New SqlCommand

    oComando.Connection = oConexion

    oComando.CommandText = "SELECT * FROM Grados"

    oAdaptador = New SqlDataAdapter(oComando)

```

```

oCmdBuilder = New SqlCommandBuilder(oAdaptador)

oDataSet = New DataSet

oConexion.Open()

oAdaptador.Fill(oDataSet, "Grados")

If oDataSet.Tables("Grados").Rows.Count = 0 Then

Else

    muestra_datos()

End If

Me.Label3.Text = "Recs. " & Me.ipos + 1 & " de " &
oDataSet.Tables("Grados").Rows.Count

Me.ButtonGuardar.Enabled = False

Me.ButtonInicio.Enabled = True

Me.ButtonAnterior.Enabled = True

Me.ButtonNuevo.Enabled = True

Me.ButtonBorrar.Enabled = True

Me.ButtonSiguiete.Enabled = True

Me.ButtonUltimo.Enabled = True

Me.ButtonEditar.Enabled = True

Me.ButtonDeshacer.Enabled = True

Me.TextBox1.ReadOnly = True

Me.TextBox2.ReadOnly = True

Me.TextBox3.ReadOnly = True

agregando = False

editando = False

End Sub

Public Sub muestra_datos()

    odatarow = oDataSet.Tables("Grados").Rows(ipos)

```

```
Me.TextBox1.Text = odatarow("IdGrado")  
Me.TextBox2.Text = odatarow("NombreGrado")  
Me.TextBox3.Text = odatarow("nomenclatura")  
  
End Sub
```

BOTON INICIO

```
Private Sub ButtonInicio_Click(ByVal sender As System.Object, ByVal  
e As System.EventArgs) Handles ButtonInicio.Click  
  
    ipos = 0  
  
    Me.muestra_datos()  
  
    Me.Label3.Text = "Recs. " & Me.ipos + 1 & " de " &  
oDataSet.Tables("Grados").Rows.Count  
  
End Sub
```

BOTON SIGUIENTE

```
Private Sub ButtonSiguiente_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles ButtonSiguiente.Click  
  
    If ipos = oDataSet.Tables("Grados").Rows.Count - 1 Then  
  
        MessageBox.Show("Ult. rec.")  
  
    Else  
  
        ipos += 1  
  
        Me.muestra_datos()  
  
    End If  
  
    Me.Label3.Text = "Recs. " & Me.ipos + 1 & " de " &  
oDataSet.Tables("Grados").Rows.Count  
  
End Sub
```

BOTON ANTERIOR

```
Private Sub ButtonAnterior_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles ButtonAnterior.Click
```

```
    If ipos = 0 Then
```

```
        MessageBox.Show("Pri. rec.")
```

```
    Else
```

```
        ipos -= 1
```

```
        Me.muestra_datos()
```

```
    End If
```

```
    Me.Label3.Text = "Recs. " & Me.ipos + 1 & " de " &  
oDataSet.Tables("Grados").Rows.Count
```

```
End Sub
```

BOTON GUARDAR

```
Private Sub ButtonGuardar_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles ButtonGuardar.Click
```

```
    Dim nuevo As New classGrados
```

```
    If agregando Then
```

```
        ' se instancia la variable nuevo al sub agregar
```

```
        nuevo.agregar_grado(Me.TextBox1.Text, Me.TextBox2.Text)
```

```
    End If
```

```
    If editando Then
```

```
        Dim editar As New classGrados
```

```
        editar.modi_grado(Me.TextBox1.Text, Me.TextBox2.Text, ipos)
```

```
    End If
```

```
refrescar()
```

```
Me.ButtonGuardar.Enabled = False  
Me.ButtonInicio.Enabled = True  
Me.ButtonAnterior.Enabled = True  
Me.ButtonNuevo.Enabled = True  
Me.ButtonBorrar.Enabled = True  
Me.ButtonSiguiete.Enabled = True  
Me.ButtonUltimo.Enabled = True  
Me.ButtonEditar.Enabled = True  
Me.ButtonDeshacer.Enabled = True  
Me.TextBox1.ReadOnly = True  
Me.TextBox2.ReadOnly = True  
Me.TextBox3.ReadOnly = True  
agregando = False
```

```
editando = False
```

```
End Sub
```

BOTON ULTIMO

```
Private Sub ButtonUltimo_Click(ByVal sender As System.Object, ByVal  
e As System.EventArgs) Handles ButtonUltimo.Click
```

```
    ipos = oDataSet.Tables("Grados").Rows.Count - 1
```

```
    Me.muestra_datos()
```

```
    Me.Label3.Text = "Recs. " & Me.ipos + 1 & " de " &  
oDataSet.Tables("Grados").Rows.Count
```

```
End Sub
```

BOTON NUEVO

```
Private Sub ButtonNuevo_Click(ByVal sender As System.Object, ByVal  
e As System.EventArgs) Handles ButtonNuevo.Click
```

```
    Me.TextBox1.ReadOnly = False  
    Me.TextBox2.ReadOnly = False  
    Me.TextBox3.ReadOnly = False  
    Me.ButtonGuardar.Enabled = True  
    Me.ButtonInicio.Enabled = False  
    Me.ButtonAnterior.Enabled = False  
    Me.ButtonNuevo.Enabled = False  
    Me.ButtonBorrar.Enabled = False  
    Me.ButtonSiguiete.Enabled = False  
    Me.ButtonUltimo.Enabled = False  
    Me.ButtonEditar.Enabled = False  
    Me.ButtonDeshacer.Enabled = True  
    agregando = True  
    Me.TextBox1.Text = ""  
    Me.TextBox2.Text = ""  
    Me.TextBox3.Text = ""  
    Me.TextBox1.Focus()
```

```
End Sub
```

BOTON EDITAR

```
Private Sub ButtonEditor_Click(ByVal sender As System.Object, ByVal  
e As System.EventArgs) Handles ButtonEditor.Click
```

```
    Me.TextBox2.ReadOnly = False  
  
    Me.TextBox3.ReadOnly = False  
  
    Me.ButtonGuardar.Enabled = True  
  
    Me.ButtonInicio.Enabled = False  
  
    Me.ButtonAnterior.Enabled = False  
  
    Me.ButtonNuevo.Enabled = False  
  
    Me.ButtonBorrar.Enabled = False  
  
    Me.ButtonSiguiente.Enabled = False  
  
    Me.ButtonUltimo.Enabled = False  
  
    Me.ButtonEditar.Enabled = False  
  
    Me.ButtonDeshacer.Enabled = True  
  
    editando = True  
  
    Me.TextBox2.Focus()
```

```
End Sub
```

BOTON BORRAR

```
Private Sub ButtonBorrar_Click(ByVal sender As System.Object, ByVal  
e As System.EventArgs) Handles ButtonBorrar.Click
```

```
    Dim borrar As String  
  
    borrar = MsgBox("Desea eliminar el registro", MsgBoxStyle.YesNo  
+ MsgBoxStyle.Question, "Advertencia")
```



```

If borrar = MsgBoxResult.Yes Then
    Dim eliminar As New ClassGrados
    eliminar.borrar_grado(ipos)
    refrescar()

End If

End Sub

Private Sub refrescar()
    'codigo para refrescar el grid con el nuevo item agregado
    oConexion = New SqlConnection("Server=(local)\SQLEXPRESS;" & _
    "DataBase=colegioanglicano;" & _
    "Integrated Security=SSPI;Connect Timeout=5")
    oComando = New SqlCommand
    oComando.Connection = oConexion
    oComando.CommandText = "SELECT * FROM Grados"
    oAdaptador = New SqlDataAdapter(oComando)
    oCmdBuilder = New SqlCommandBuilder(oAdaptador)
    oDataSet = New DataSet
    oConexion.Open()
    oAdaptador.Fill(oDataSet, "Grados")

    If oDataSet.Tables("Grados").Rows.Count = 0 Then
    Else
        ipos = 0
        odatarow = oDataSet.Tables("Grados").Rows(ipos)
        Me.muestra_datos()
    End If
End Sub

```

```
End If

Me.Label3.Text = "Recs. " & Me.ipos + 1 & " de " &
oDataSet.Tables("Grados").Rows.Count

End Sub
```

BOTON DESHACER

```
Private Sub ButtonDeshacer_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ButtonDeshacer.Click

Me.ButtonGuardar.Enabled = False

Me.ButtonInicio.Enabled = True

Me.ButtonAnterior.Enabled = True

Me.ButtonNuevo.Enabled = True

Me.ButtonBorrar.Enabled = True

Me.ButtonSiguiente.Enabled = True

Me.ButtonUltimo.Enabled = True

Me.ButtonEditar.Enabled = True

Me.ButtonDeshacer.Enabled = True

Me.TextBox1.ReadOnly = True

Me.TextBox2.ReadOnly = True

Me.TextBox3.ReadOnly = True

refrescar()

End Sub
```

```
Private Sub TextBox2_GotFocus(ByVal sender As Object, ByVal e As
System.EventArgs) Handles TextBox2.GotFocus

If agregando Then
```

```

oConexion = New SqlConnection("Server=(local)\SQLEXPRESS;"
& _
"DataBase=colegioanglicano;" & _
"Integrated Security=SSPI;Connect Timeout=5")
oComando = New SqlCommand
oComando.Connection = oConexion
oComando.CommandText = "SELECT * FROM Grados where idgrado
= '" & Me.TextBox1.Text & "';"
oAdaptador = New SqlDataAdapter(oComando)
oCmdBuilder = New SqlCommandBuilder(oAdaptador)
oDataSet = New DataSet
oConexion.Open()
oAdaptador.Fill(oDataSet, "Grados")
If oDataSet.Tables("Grados").Rows.Count > 0 Then
    Me.TextBox1.Focus()
    MessageBox.Show("Codigo ya existe", "*Advertencia*",
MessageBoxButtons.OK, MessageBoxIcon.Warning)
End If
oConexion = New SqlConnection("Server=(local)\SQLEXPRESS;"
& _
"DataBase=colegioanglicano;" & _
"Integrated Security=SSPI;Connect Timeout=5")
oComando = New SqlCommand
oComando.Connection = oConexion
oComando.CommandText = "SELECT * FROM Grados;"
oAdaptador = New SqlDataAdapter(oComando)
oCmdBuilder = New SqlCommandBuilder(oAdaptador)
oDataSet = New DataSet
oConexion.Open()

```

```
        oAdaptador.Fill(oDataSet, "Grados")
    End If
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    vista.ShowDialog()
End Sub
End Class
```

EN LA BARRA DE HERRAMIENTAS DEL MENU PRINCIPAL EN CATALOGOS
CODIGO PARA CARGAR EL FORMULARIO PROFESORES EN EL MENU
PRINCIPAL

```
Private Sub ProfesoresToolStripMenuItem_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
ProfesoresToolStripMenuItem.Click
```

```
    frmprofesores.ShowDialog()
```

```
End Sub
```

The image shows a screenshot of a Windows application window titled "Catalogo de Profesores". The window contains a form with the following fields and controls:

- Id**: A small text input field.
- Nombres**: A text input field.
- Apellidos**: A text input field.
- Sexo**: A checkbox.
- Dirección**: A wide text input field.
- Telefono**: A text input field.
- Fec. Nac.**: A date input field with slashes for separators.
- Cedula**: A text input field with dashes for separators.
- INSS**: A text input field.
- Guia**: A checkbox.
- Barra de Botones**: A toolbar containing icons for back, forward, search, and other navigation functions.
- Label11**: A label at the bottom of the form.

CODIGO FORMULARIO PROFESORES

```
Imports System.Data
```

```
Imports System.Data.SqlClient
```

```

Public Class frmprofesores

    Dim FILAS() As DataRow

    Dim oDataRow As DataRow

    Private ipos As Integer = 0

    Dim oDataSet As DataSet

    Dim oAdaptador As SqlDataAdapter

    Dim agregando As Boolean, editando As Boolean, undo As Boolean

    Dim oCmdBuilder As SqlCommandBuilder

    Dim oComando As SqlCommand

    Dim oConexion As New SqlConnection

    Private Sub frmprofesores_FormClosing(ByVal sender As Object, ByVal
e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

        Dim Resp As MsgBoxResult

        If agregando Then

            Resp = MsgBox("Desea salir", MsgBoxStyle.YesNo)

            If Resp = MsgBoxResult.Yes Then

                agregando = False

                e.Cancel = False

                Me.Close()

            Else

                e.Cancel = True

            End If

        End If

        If editando Then

            Resp = MsgBox("Desea salir", MsgBoxStyle.YesNo)

```

```

        If Resp = MsgBoxResult.Yes Then
            editando = False
            e.Cancel = False
            Me.Close()
        Else
            e.Cancel = True
        End If
    End If

End Sub

```

```

Private Sub frmprofesores_Load(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles MyBase.Load

```

```

'este codigo es para actualizar el grid
oConexion = New SqlConnection("Server=(local)\SQLEXPRESS;" & _
"DataBase=colegioanglicano;" & _
"Integrated Security=SSPI;Connect Timeout=5")
oComando = New SqlCommand
oComando.Connection = oConexion
oComando.CommandText = "SELECT * FROM profesores"
oAdaptador = New SqlDataAdapter(oComando)
oCmdBuilder = New SqlCommandBuilder(oAdaptador)
oDataSet = New DataSet
oConexion.Open()
oAdaptador.Fill(oDataSet, "Profesores")
If oDataSet.Tables("profesores").Rows.Count = 0 Then

```

```

Else

    odatarow = oDataSet.Tables("Profesores").Rows(ipos)

    Me.TextBox1.Text = odatarow("Idprofesor")

    Me.TextBox2.Text = odatarow("Nombres")

    Me.TextBox3.Text = odatarow("apellidos")

    Me.TextBox4.Text = odatarow("sexo")

    Me.TextBox5.Text = odatarow("direccion")

    Me.TextBox6.Text = odatarow("telefono")

    Me.MaskedTextBox1.Text = odatarow("fecha_nac")

    Me.MaskedTextBox2.Text = odatarow("cedula")

    Me.TextBox7.Text = odatarow("inss")

    If odatarow("prof_guia") = 1 Then

        Me.CheckBox1.Checked = True

    Else

        Me.CheckBox1.Checked = False

    End If

    Me.Label11.Text = "Recs. " & Me.ipos + 1 & " de " &
oDataSet.Tables("profesores").Rows.Count

End If

Me.ButtonGuardar.Enabled = False

Me.ButtonInicio.Enabled = True

Me.ButtonAnterior.Enabled = True

Me.ButtonNuevo.Enabled = True

Me.ButtonBorrar.Enabled = True

Me.ButtonSiguiente.Enabled = True

Me.ButtonUltimo.Enabled = True

Me.ButtonEditar.Enabled = True

```



```
Me.ButtonDeshacer.Enabled = False

Me.TextBox1.ReadOnly = True
Me.TextBox2.ReadOnly = True
Me.TextBox3.ReadOnly = True
Me.TextBox4.ReadOnly = True
Me.TextBox5.ReadOnly = True
Me.TextBox6.ReadOnly = True
Me.MaskedTextBox1.ReadOnly = True
Me.MaskedTextBox2.ReadOnly = True
Me.TextBox7.ReadOnly = True
Me.CheckBox1.Enabled = False

agregando = False

editando = False

End Sub
```

BOTON INICIO

```
Private Sub ButtonInicio_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonInicio.Click

    ipos = 0

    Me.muestra_datos()

End Sub
```

BOTON SIGUIENTE

```
Private Sub ButtonSiguiente_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonSiguiente.Click
```

```
        If ipos = oDataSet.Tables("Profesores").Rows.Count - 1 Then
            MessageBox.Show("Ult. rec.")
        Else
            ipos += 1
            Me.muestra_datos()

        End If
    End Sub
```

BOTON ANTERIOR

```
    Private Sub ButtonAnterior_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles ButtonAnterior.Click
        If ipos = 0 Then
            MessageBox.Show("Pri. rec.")
        Else
            ipos -= 1
            Me.muestra_datos()

        End If
    End Sub
```

BOTON GUARDAR

```
    Private Sub ButtonGuardar_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles ButtonGuardar.Click
        Dim nuevo As New ClassProfesores
        If agregando Then
            ' se instancia la variable nuevo al sub agregar
```

```
nuevo.agregar_Profesores()  
  
End If  
  
If editando Then  
  
    Dim editar As New ClassProfesores  
  
    editar.MODIFICAR_profesores(Me.TextBox1.Text,  
Me.TextBox2.Text, ipos)  
  
End If  
  
refrescar()  
  
Me.ButtonGuardar.Enabled = False  
Me.ButtonInicio.Enabled = True  
Me.ButtonAnterior.Enabled = True  
Me.ButtonNuevo.Enabled = True  
Me.ButtonBorrar.Enabled = True  
Me.ButtonSiguiente.Enabled = True  
Me.ButtonUltimo.Enabled = True  
Me.ButtonEditar.Enabled = True  
Me.ButtonDeshacer.Enabled = False  
  
Me.TextBox1.ReadOnly = True  
Me.TextBox2.ReadOnly = True  
Me.TextBox3.ReadOnly = True  
Me.TextBox4.ReadOnly = True  
Me.TextBox5.ReadOnly = True  
Me.TextBox6.ReadOnly = True  
Me.MaskedTextBox1.ReadOnly = True  
Me.MaskedTextBox2.ReadOnly = True  
Me.TextBox7.ReadOnly = True
```

```
Me.CheckBox1.Enabled = False
```

```
agregando = False
```

```
editando = False
```

```
End Sub
```

BOTON ULTIMO

```
Private Sub ButtonUltimo_Click(ByVal sender As System.Object, ByVal  
e As System.EventArgs) Handles ButtonUltimo.Click
```

```
    ipos = oDataSet.Tables("Profesores").Rows.Count - 1
```

```
    Me.muestra_datos()
```

```
End Sub
```

BOTON NUEVO

```
Private Sub ButtonNuevo_Click(ByVal sender As System.Object, ByVal  
e As System.EventArgs) Handles ButtonNuevo.Click
```

```
    Me.TextBox1.ReadOnly = False
```

```
    Me.TextBox2.ReadOnly = False
```

```
    Me.TextBox3.ReadOnly = False
```

```
    Me.TextBox4.ReadOnly = False
```

```
    Me.TextBox5.ReadOnly = False
```

```
    Me.TextBox6.ReadOnly = False
```

```
Me.MaskedTextBox1.ReadOnly = False
Me.MaskedTextBox2.ReadOnly = False
Me.TextBox7.ReadOnly = False
Me.CheckBox1.Enabled = True
```

```
Me.ButtonGuardar.Enabled = True
Me.ButtonInicio.Enabled = False
Me.ButtonAnterior.Enabled = False
Me.ButtonNuevo.Enabled = False
Me.ButtonBorrar.Enabled = False
Me.ButtonSiguiente.Enabled = False
Me.ButtonUltimo.Enabled = False
Me.ButtonEditar.Enabled = False
Me.ButtonDeshacer.Enabled = True
agregando = True
```

```
Me.TextBox1.Text = ""
```

```
Me.TextBox2.Text = ""
```

```
Me.TextBox3.Text = ""
```

```
Me.TextBox4.Text = ""
```

```
Me.TextBox5.Text = ""
```

```
Me.TextBox6.Text = ""
```

```
Me.MaskedTextBox1.Text = ""
```

```
Me.MaskedTextBox2.Text = ""
```

```
Me.TextBox7.Text = ""
```

```
Me.CheckBox1.Checked = False
```

```
Me.TextBox1.Focus()
```

```
End Sub
```

BOTON EDITAR

```
Private Sub ButtonEditor_Click(ByVal sender As System.Object, ByVal  
e As System.EventArgs) Handles ButtonEditor.Click
```

```
    Me.TextBox2.ReadOnly = False  
    Me.TextBox3.ReadOnly = False  
    Me.TextBox4.ReadOnly = False  
    Me.TextBox5.ReadOnly = False  
    Me.TextBox6.ReadOnly = False  
    Me.MaskedTextBox1.ReadOnly = False  
    Me.MaskedTextBox2.ReadOnly = False  
    Me.TextBox7.ReadOnly = False  
    Me.CheckBox1.Enabled = True
```

```
    Me.ButtonGuardar.Enabled = True  
    Me.ButtonInicio.Enabled = False  
    Me.ButtonAnterior.Enabled = False  
    Me.ButtonNuevo.Enabled = False  
    Me.ButtonBorrar.Enabled = False  
    Me.ButtonSiguiente.Enabled = False  
    Me.ButtonUltimo.Enabled = False  
    Me.ButtonEditor.Enabled = False  
    Me.ButtonDeshacer.Enabled = True  
    editando = True  
    Me.TextBox2.Focus()
```

```
End Sub
```

BOTON BORRAR

```
Private Sub ButtonBorrar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonBorrar.Click
```

```
    Dim borrar As String
```

```
    borrar = MsgBox("Desea eliminar el registro", MsgBoxStyle.YesNo + MsgBoxStyle.Question, "Advertencia")
```

```
    If borrar = MsgBoxResult.Yes Then
```

```
        Dim eliminar As New ClassProfesores
```

```
        eliminar.borrar_profesores(ipos)
```

```
        refrescar()
```

```
    End If
```

```
End Sub
```

BOTON DESHACER

```
Private Sub ButtonDeshacer_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonDeshacer.Click
```

```
    agregando = False
```

```
    editando = False
```

```
    undo = True
```

```
    Me.ButtonGuardar.Enabled = False
```

```
    Me.ButtonInicio.Enabled = True
```

```
    Me.ButtonAnterior.Enabled = True
```

```
Me.ButtonNuevo.Enabled = True
Me.ButtonBorrar.Enabled = True
Me.ButtonSiguiete.Enabled = True
Me.ButtonUltimo.Enabled = True
Me.ButtonEditar.Enabled = True
Me.ButtonDeshacer.Enabled = True
```

```
Me.TextBox1.ReadOnly = True
Me.TextBox2.ReadOnly = True
Me.TextBox3.ReadOnly = True
Me.TextBox4.ReadOnly = True
Me.TextBox5.ReadOnly = True
Me.TextBox6.ReadOnly = True
Me.MaskedTextBox1.ReadOnly = True
Me.MaskedTextBox2.ReadOnly = True
Me.TextBox7.ReadOnly = True
Me.CheckBox1.Enabled = False
```

```
muestra_datos()
```

```
End Sub
```

```
Public Sub muestra_datos()
```

```
odatarow = oDataSet.Tables("Profesores").Rows(ipos)
```

```
Me.TextBox1.Text = odatarow("Idprofesor")
```

```
Me.TextBox2.Text = odatarow("Nombres")
```

```
Me.TextBox3.Text = odatarow("apellidos")
```

```
Me.TextBox4.Text = odatarow("sexo")
```



```

Me.TextBox5.Text = odatarow("direccion")
Me.TextBox6.Text = odatarow("telefono")
Me.MaskedTextBox1.Text = odatarow("fecha_nac")
Me.MaskedTextBox2.Text = odatarow("cedula")
Me.TextBox7.Text = odatarow("inss")
If odatarow("prof_guia") = 1 Then
    Me.CheckBox1.Checked = True
Else
    Me.CheckBox1.Checked = False
End If
End Sub

Private Sub ButtonDeshacer_MouseMove(ByVal sender As Object, ByVal
e As System.Windows.Forms.MouseEventArgs) Handles
ButtonDeshacer.MouseMove
    undo = True
End Sub

Private Sub refrescar()
    'codigo para refrescar el grid con el nuevo item agregado
    oConexion = New SqlConnection("Server=(local)\SQLEXPRESS;" & _
    "DataBase=colegioanglicano;" & _
    "Integrated Security=SSPI;Connect Timeout=5")
    oComando = New SqlCommand
    oComando.Connection = oConexion
    oComando.CommandText = "SELECT * FROM profesores"
    oAdaptador = New SqlDataAdapter(oComando)
    oCmdBuilder = New SqlCommandBuilder(oAdaptador)
    oDataSet = New DataSet
    oConexion.Open()

```

```

oAdaptador.Fill(oDataSet, "profesores")

ipos = 0

odatarow = oDataSet.Tables("profesores").Rows(ipos)

Me.muestra_datos()

Me.Label13.Text = "Recs. " & Me.ipos + 1 & " de " &
oDataSet.Tables("profesores").Rows.Count

End Sub

Private Sub TextBox2_GotFocus(ByVal sender As Object, ByVal e As
System.EventArgs) Handles TextBox2.GotFocus

    If agregando Then

        oConexion = New SqlConnection("Server=(local)\SQLEXPRESS;"
& _
        "DataBase=colegioanglicano;" & _
        "Integrated Security=SSPI;Connect Timeout=5")

        oComando = New SqlCommand

        oComando.Connection = oConexion

        oComando.CommandText = "SELECT * FROM profesores where
idprofesor = '" & Me.TextBox1.Text & "';"

        oAdaptador = New SqlDataAdapter(oComando)

        oCmdBuilder = New SqlCommandBuilder(oAdaptador)

        oDataSet = New DataSet

        oConexion.Open()

        oAdaptador.Fill(oDataSet, "Profesores")

        If oDataSet.Tables("Profesores").Rows.Count > 0 Then

            Me.TextBox1.Focus()

            MessageBox.Show("Codigo ya existe", "*Advertencia*",
MessageBoxButtons.OK, MessageBoxIcon.Warning)

        End If

```

```

oConexion = New SqlConnection("Server=(local)\SQLEXPRESS;"
& _
"DataBase=colegioanglicano;" & _
"Integrated Security=SSPI;Connect Timeout=5")
oComando = New SqlCommand
oComando.Connection = oConexion
oComando.CommandText = "SELECT * FROM Profesores;"
oAdaptador = New SqlDataAdapter(oComando)
oCmdBuilder = New SqlCommandBuilder(oAdaptador)
oDataSet = New DataSet
oConexion.Open()
oAdaptador.Fill(oDataSet, "Profesores")

End If

End Sub

Private Sub TextBox4_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TextBox4.TextChanged

End Sub

End Class

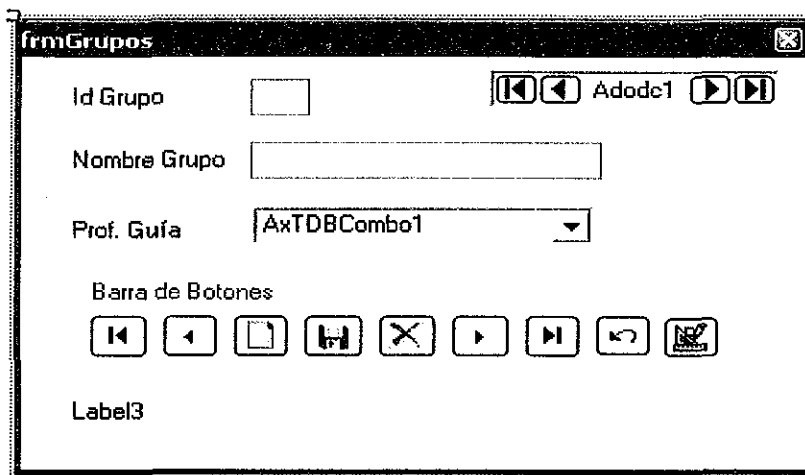
```

CODIGO PARA CARGAR EL FORMULARIO GRUPOS EN EL MENU PRINCIPAL

```
Private Sub GruposToolStripMenuItem_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
GruposToolStripMenuItem.Click
```

```
    frmGrupos.ShowDialog()
```

```
    frmGrupos.Close()
```



CODIGO FORMULARIO GRUPOS

```
Imports System.Data
```

```
Imports System.Data.SqlClient
```

```
Public Class frmGrupos
```

```
    Dim oDataRow As DataRow
```

```
    Private ipos As Integer = 0
```

```
    Dim oDataSet As DataSet
```

```
    Dim oAdaptador As SqlDataAdapter
```

```
    Dim agregando As Boolean, editando As Boolean
```

```

Dim oCmdBuilder As SqlCommandBuilder

Dim oComando As SqlCommand

Dim oConexion As New SqlConnection

Private Sub frmGrupos_FormClosing(ByVal sender As Object, ByVal e
As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

    If agregando Then

        Dim Resp As MsgBoxResult

        Resp = MsgBox("Desea salir", MsgBoxStyle.YesNo)

        If Resp = MsgBoxResult.Yes Then

            agregando = False

            e.Cancel = False

            Me.Close()

        Else

            e.Cancel = True

        End If

    End If

    If editando Then

        Dim Resp As MsgBoxResult

        Resp = MsgBox("Desea salir", MsgBoxStyle.YesNo)

        If Resp = MsgBoxResult.Yes Then

            editando = False

            Me.Close()

            e.Cancel = False

```

```

        Else

            e.Cancel = True

        End If

    End If

End Sub

Private Sub frmGrupos_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load

    'este codigo es para actualizar el grid

    oConexion = New SqlConnection("Server=(local)\SQLEXPRESS;" & _
    "DataBase=colegioanglicano;" & _
    "Integrated Security=SSPI;Connect Timeout=5")

    oComando = New SqlCommand

    oComando.Connection = oConexion

    oComando.CommandText = "SELECT * FROM Grupos"

    oAdaptador = New SqlDataAdapter(oComando)

    oCmdBuilder = New SqlCommandBuilder(oAdaptador)

    oDataSet = New DataSet

    oConexion.Open()

    oAdaptador.Fill(oDataSet, "Grupos")

    If oDataSet.Tables("Grupos").Rows.Count = 0 Then

        Else

            muestra_datos()

        End If

        Me.Label3.Text = "Recs. " & Me.ipos + 1 & " de " &
oDataSet.Tables("Grupos").Rows.Count

        Me.ButtonGuardar.Enabled = False

```

```

Me.ButtonInicio.Enabled = True
Me.ButtonAnterior.Enabled = True
Me.ButtonNuevo.Enabled = True
Me.ButtonBorrar.Enabled = True
Me.ButtonSiguiente.Enabled = True
Me.ButtonUltimo.Enabled = True
Me.ButtonEditar.Enabled = True
Me.ButtonDeshacer.Enabled = True
Me.TextBox1.ReadOnly = True
Me.TextBox2.ReadOnly = True
Me.AxTDBComb1.Enabled = False
agregando = False
editando = False

End Sub

Public Sub muestra_datos()
    odatarow = oDataSet.Tables("Grupos").Rows(ipos)
    Me.TextBox1.Text = odatarow("IdGrupo")
    Me.TextBox2.Text = odatarow("NombreGrupo")
    Me.AxTDBComb1.BoundText = odatarow("idprofesor")

End Sub

```

BOTON INICIO

```

Private Sub ButtonInicio_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ButtonInicio.Click
    ipos = 0
    Me.muestra_datos()

```

```
Me.Label3.Text = "Recs. " & Me.ipos + 1 & " de " &  
oDataSet.Tables("Grupos").Rows.Coun
```

```
End Sub
```

BOTON SIGUIENTE

```
Private Sub ButtonSiguiente_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles ButtonSiguiente.Click
```

```
    If ipos = oDataSet.Tables("Grupos").Rows.Count - 1 Then
```

```
        MessageBox.Show("Ult. rec.")
```

```
    Else
```

```
        ipos += 1
```

```
        Me.muestra_datos()
```

```
    End If
```

```
    Me.Label3.Text = "Recs. " & Me.ipos + 1 & " de " &  
oDataSet.Tables("Grupos").Rows.Count
```

```
End Sub
```

BOTON ANTERIOR

```
Private Sub ButtonAnterior_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles ButtonAnterior.Click
```

```
    If ipos = 0 Then
```

```
        MessageBox.Show("Pri. rec.")
```

```
    Else
```

```
        ipos -= 1
```

```
        Me.muestra_datos()
```

```
    End If
```

```
    Me.Label3.Text = "Recs. " & Me.ipos + 1 & " de " &  
oDataSet.Tables("Grupos").Rows.Count
```

```
End Sub
```


BOTON GUARDAR

```
Private Sub ButtonGuardar_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles ButtonGuardar.Click
```

```
    Dim nuevo As New classGrupos
```

```
    If agregando Then
```

```
        ' se instancia la variable nuevo al sub agregar
```

```
        nuevo.agregar_grupos()
```

```
    End If
```

```
    If editando Then
```

```
        Dim editar As New classGrupos
```

```
        editar.modi_grupos(ipos)
```

```
    End If
```

```
    refrescar()
```

```
    Me.ButtonGuardar.Enabled = False
```

```
    Me.ButtonInicio.Enabled = True
```

```
    Me.ButtonAnterior.Enabled = True
```

```
    Me.ButtonNuevo.Enabled = True
```

```
    Me.ButtonBorrar.Enabled = True
```

```
    Me.ButtonSiguiente.Enabled = True
```

```
    Me.ButtonUltimo.Enabled = True
```

```
    Me.ButtonEditar.Enabled = True
```

```
    Me.ButtonDeshacer.Enabled = True
```

```
    Me.TextBox1.ReadOnly = True
```

```
    Me.TextBox2.ReadOnly = True
```

```
    Me.AxTDBCombo1.Enabled = False
```

```
    agregando = False
    editando = False
End Sub
```

BOTON ULTIMO

```
Private Sub ButtonUltimo_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonUltimo.Click
    ipos = oDataSet.Tables("Grupos").Rows.Count - 1
    Me.muestra_datos()
    Me.Label3.Text = "Recs. " & Me.ipos + 1 & " de " & oDataSet.Tables("Grupos").Rows.Count
End Sub
```

BOTON NUEVO

```
Private Sub ButtonNuevo_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonNuevo.Click
    Me.TextBox1.ReadOnly = False
    Me.TextBox2.ReadOnly = False
    Me.AxTDBComb1.Enabled = True

    Me.ButtonGuardar.Enabled = True
    Me.ButtonInicio.Enabled = False
    Me.ButtonAnterior.Enabled = False
    Me.ButtonNuevo.Enabled = False
    Me.ButtonBorrar.Enabled = False
    Me.ButtonSiguiente.Enabled = False
```

```
Me.ButtonUltimo.Enabled = False
Me.ButtonEditar.Enabled = False
Me.ButtonDeshacer.Enabled = True
agregando = True
Me.TextBox1.Text = ""
Me.TextBox2.Text = ""

Me.TextBox1.Focus()
```

End Sub

BOTON EDITAR

```
Private Sub ButtonEditar_Click(ByVal sender As System.Object, ByVal  
e As System.EventArgs) Handles ButtonEditar.Click
```

```
Me.TextBox2.ReadOnly = False
Me.AxTDBCombol.Enabled = True
Me.ButtonGuardar.Enabled = True
Me.ButtonInicio.Enabled = False
Me.ButtonAnterior.Enabled = False
Me.ButtonNuevo.Enabled = False
Me.ButtonBorrar.Enabled = False
```

```
Me.ButtonSiguiente.Enabled = False
Me.ButtonUltimo.Enabled = False
Me.ButtonEditar.Enabled = False
Me.ButtonDeshacer.Enabled = True
editando = True
Me.TextBox2.Focus()
```

```
End Sub
```

BOTON BORRAR

```
Private Sub ButtonBorrar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonBorrar.Click
```

```
Dim borrar As String
```

```
borrar = MsgBox("Desea eliminar el registro", MsgBoxStyle.YesNo + MsgBoxStyle.Question, "Advertencia")
```

```
If borrar = MsgBoxResult.Yes Then
```

```
Dim eliminar As New ClassGrupos
```

```
eliminar.borrar_grupos(ipos)
```

```
refrescar()
```

```
End If
```

```
End Sub
```

```
Private Sub refrescar()
```

```
'codigo para refrescar el grid con el nuevo item agregado
```

```
oConexion = New SqlConnection("Server=(local)\SQLEXPRESS;" & _
```

```
"DataBase=colegioanglicano;" & _
```

```
"Integrated Security=SSPI;Connect Timeout=5")
```

```

oComando = New SqlCommand
oComando.Connection = oConexion
oComando.CommandText = "SELECT * FROM Grupos"
oAdaptador = New SqlDataAdapter(oComando)
oCmdBuilder = New SqlCommandBuilder(oAdaptador)
oDataSet = New DataSet
oConexion.Open()
oAdaptador.Fill(oDataSet, "Grupos")

If oDataSet.Tables("Grupos").Rows.Count = 0 Then
Else
    ipos = 0
    odatarow = oDataSet.Tables("Grupos").Rows(ipos)
    Me.muestra_datos()
End If

Me.Label3.Text = "Recs. " & Me.ipos + 1 & " de " &
oDataSet.Tables("Grupos").Rows.Count

End Sub

```

BOTON DESHACER

```

Private Sub ButtonDeshacer_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ButtonDeshacer.Click

Me.ButtonGuardar.Enabled = False

Me.ButtonInicio.Enabled = True

Me.ButtonAnterior.Enabled = True

Me.ButtonNuevo.Enabled = True

Me.ButtonBorrar.Enabled = True

Me.ButtonSiguiente.Enabled = True

```

```

Me.ButtonUltimo.Enabled = True

Me.ButtonEditar.Enabled = True

Me.ButtonDeshacer.Enabled = True

Me.TextBox1.ReadOnly = True

Me.TextBox2.ReadOnly = True

Me.AxTDBCombo1.Enabled = False

refrescar()

End Sub

Private Sub TextBox2_GotFocus(ByVal sender As Object, ByVal e As
System.EventArgs) Handles TextBox2.GotFocus

    If agregando Then

        oConexion = New SqlConnection("Server=(local)\SQLEXPRESS;"
& _
        "DataBase=colegioanglicano;" & _
        "Integrated Security=SSPI;Connect Timeout=5")

        oComando = New SqlCommand

        oComando.Connection = oConexion

        oComando.CommandText = "SELECT * FROM Grupos where idgrupo
= '" & Me.TextBox1.Text & "';"

        oAdaptador = New SqlDataAdapter(oComando)

        oCmdBuilder = New SqlCommandBuilder(oAdaptador)

        oDataSet = New DataSet

        oConexion.Open()

        oAdaptador.Fill(oDataSet, "Grupos")

        If oDataSet.Tables("Grupos").Rows.Count > 0 Then

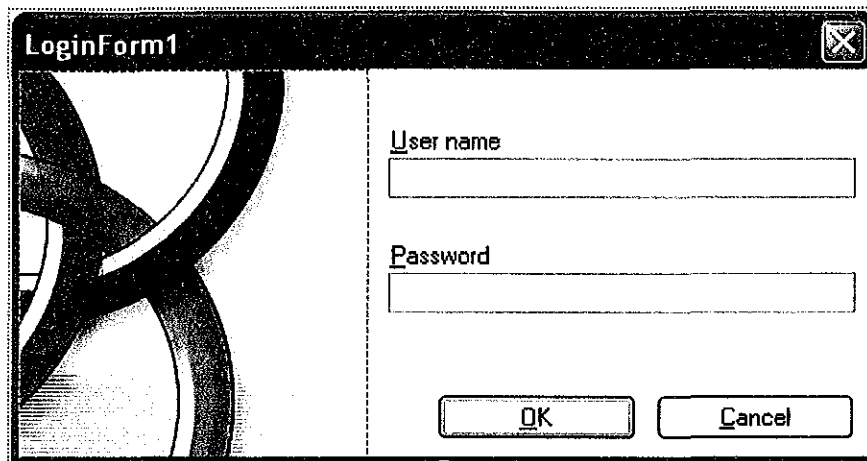
            Me.TextBox1.Focus()

            MessageBox.Show("Codigo ya existe", "*Advertencia*",
MessageBoxButtons.OK, MessageBoxIcon.Warning)

```

```
End If
oConexion = New SqlConnection("Server=(local)\SQLEXPRESS;"
& _
"DataBase=colegioanglicano;" & _
"Integrated Security=SSPI;Connect Timeout=5")
oComando = New SqlCommand
oComando.Connection = oConexion
oComando.CommandText = "SELECT * FROM Grupos;"
oAdaptador = New SqlDataAdapter(oComando)
oCmdBuilder = New SqlCommandBuilder(oAdaptador)
oDataSet = New DataSet
oConexion.Open()
oAdaptador.Fill(oDataSet, "Grupos")
End If
End Sub
End Class
```

LOGIN O ENTRADA AL SISTEMA



The image shows a Windows-style dialog box titled "LoginForm1". On the left side, there is a decorative graphic consisting of overlapping curved lines. On the right side, there are two text input fields. The first field is labeled "User name" and the second is labeled "Password". Below these fields are two buttons: "OK" and "Cancel".

BOTON OK

```
Private Sub OK_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles OK.Click

    oConexion = New SqlConnection("Server=(local);" & _
    "DataBase=colegioanglicano;" & _
    "Integrated Security=SSPI;Connect Timeout=5")

    oComando = New SqlCommand

    oComando.Connection = oConexion

    oComando.CommandText = "SELECT * FROM usuarios where usuario =
'" & Me.UsernameTextBox.Text & "'"

    oAdaptador = New SqlDataAdapter(oComando)

    oCmdBuilder = New SqlCommandBuilder(oAdaptador)

    oDataSet = New DataSet

    oConexion.Open()

    oAdaptador.Fill(oDataSet, "usuarios")

    If oDataSet.Tables("usuarios").Rows.Count = 0 Then
```



```

        Me.UsernameTextBox.Focus()

        oConexion.Close()

        MessageBox.Show("Usuario no existe")

        Exit Sub

    End If

    oConexion.Close()

    oComando.CommandText = "SELECT * FROM usuarios where usuario =
'" & Me.UsernameTextBox.Text & "' and password = '" &
Me.PasswordTextBox.Text & "'"

    oAdaptador = New SqlDataAdapter(oComando)

    oCmdBuilder = New SqlCommandBuilder(oAdaptador)

    oDataSet = New DataSet

    oConexion.Open()

    oAdaptador.Fill(oDataSet, "usuarios")

    If oDataSet.Tables("usuarios").Rows.Count = 0 Then

        Me.PasswordTextBox.Focus()

        oConexion.Close()

        MessageBox.Show("Password incorrecto")

        Exit Sub

    End If

    MessageBox.Show("Bienvenido " & Me.UsernameTextBox.Text)

    Me.Hide()

    MDIMain.Show()

End Sub

```

BOTON CANCEL

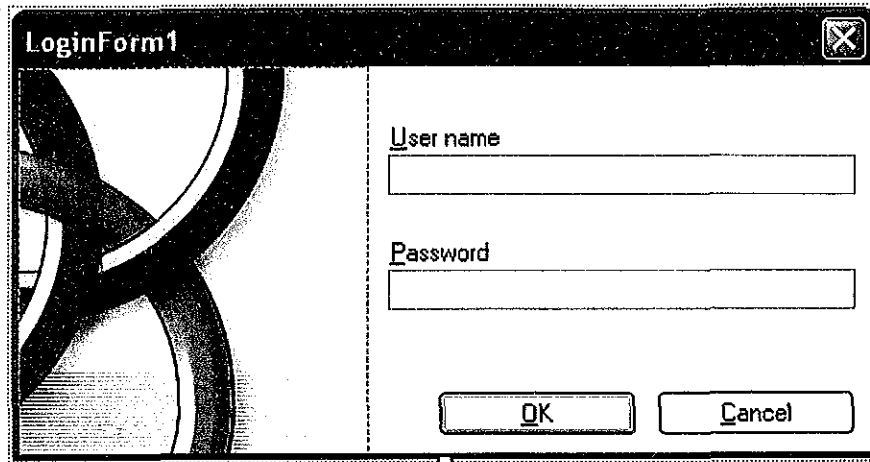
```
Private Sub Cancel_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Cancel.Click
```

```
    Me.Close()
```

```
End Sub
```

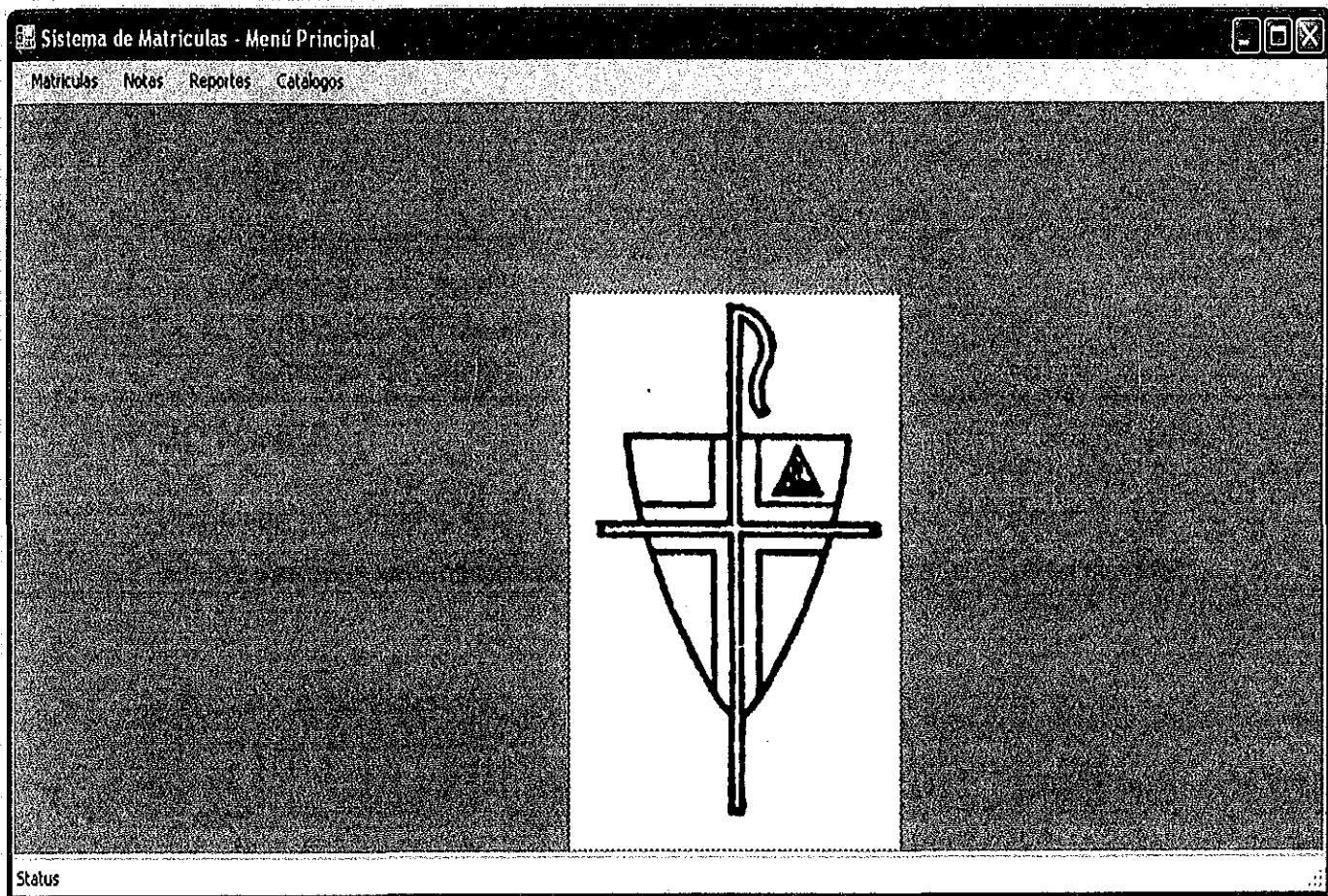
Anexo 2

MANUAL DEL USUARIO



The image shows a screenshot of a login form window titled "LoginForm1". The window has a standard Windows-style title bar with a close button (X) in the top right corner. The main area of the window is divided into two sections. On the left, there is a decorative graphic consisting of several overlapping, curved, shaded bands. On the right, there are two text input fields. The first field is labeled "User name" and the second is labeled "Password". Below these fields are two buttons: "OK" and "Cancel".

Este es el formulario login donde el usuario deberá digitar su nombre de usuario y contraseña para poder ingresar al sistema de control de matriculas y notas.



Después de haber digitado el nombre de usuario y contraseña ingresara a la pantalla de menú principal donde se encuentran en la barra de herramientas las tareas que realiza el sistema como son: Matriculas, Notas, Reportes y Catálogos. De los cuales cada uno se distribuye de la siguiente manera:

Matriculas:

- Registrar matricula.
- Asignar grupo.

Notas:

- Actualizar notas.

Reportes:

- Boletines
- Planilla

Catálogos:

- Catalogo de Materias.
- Catalogo de Grados.
- Catalogo de Profesores.
- Catalogo de Grupos.

BOTONES

Estos son los botones que nos aparecerán en las pantallas del sistema, los cuales van a determinar las acciones que vamos a realizar a la base de datos.

Barra de Botones



Primero: Este botón nos posicionará en el primer registro contenido en la base de datos.



Ultimo: Este botón nos mostrara el último registro guardado en la base de datos.



Anterior: El botón anterior nos ubica un registro antes del registro actual.



Siguiete: Nos adelanta una posición hacia arriba de los registros guardados.



Nuevo: El botón nuevo nos sirve para crear un dato nuevo de un alumno o un profesor, al dar click en el botón nos activara todas las casillas del formulario así como los demás botones de la barra de botones.



Guardar: Al dar click en este botón nos guardara en la base de datos un registro nuevo o modificado por el usuario del sistema.



Borrar: Elimina un dato agregado erróneamente o un dato que ya no será utilizado.



Deshacer: Deshace cualquier error hecho por el usuario sin tener que guardar el dato.



Modificar: Este botón sirve para realizar cambios que se necesiten hacer en algún registro contenido en la base de datos del sistema.

Registro de Alumnos - Matricula

Id Alumno

Nombres Apellidos

Direccion

Fecha_Nac Edad Sexo Telefono

Nombre Tutor

Año Lectivo C.Procedencia Repitente

Grado Turno

CLASES QUE LLEVARA EL ALUMNO

Matematicas Educacion Fisica

Español

Ciencias naturales

Inglés

Estudios Sociales

Imprimir

Barra de Botones

Recs. 1 de 10

Esta es la pantalla de registro de Alumno donde se deberán ingresar los datos del estudiante para generar la matricula utilizando los botones del formulario y posteriormente guardarla en la base de datos del sistema. Además posee la opción de imprimir un papel con los datos de la matricula.

frmAsignarGrupo

Nombres y Apellidos

Grado

Año Lectivo


Grupo

Guardar

Esta pantalla nos permite asignar un grupo al estudiante, esto va depender de la edad del alumno para así colocarlo en las distintas secciones que posee el colegio. El usuario deberá dar click en la flecha ubicada en el lado derecho de la caja de texto "Nombres y Apellidos" para desplegar la lista de los alumnos guardados en la base de datos, a continuación selecciona el alumno al cual desea asignarle el grupo, posteriormente deberá desplegar la lista de grupos en la caja de texto "Grupo" y seleccionar el grupo al que pertenecerá el alumno, después de todo este proceso finalmente el usuario deberá guardar la información presionando el botón "Guardar".



The image shows a screenshot of a Windows form titled "frmGrupos". The form contains three input fields: "Id Grupo" with the value "01", "Nombre Grupo" with the value "Primer Grado A", and "Prof. Guía" with the value "01". Below these fields is a "Barra de Botones" (toolbar) containing several icons for navigation and actions. At the bottom of the form, it displays "Recs. 1 de 23".

Este formulario se utilizara en el sistema para agregar un nuevo grupo así como modificar y borrar grupos existentes en la base de datos. Si el usuario desea crear un nuevo grupo deberá digitar el Id del grupo y luego escribir el nombre que llevará el grupo con su respectiva clasificación (A, B, C, etc). Posteriormente tendrá que desplegar la lista de profesores para asignarlo como profesor guía del grupo creado o modificado. Como paso final se guardara el nuevo dato al dar click en el botón guardar ubicado en la barra de botones.

Actualizar Notas 

Seleccione el Nombre del Estudiante

IdAlumno	Materia	1erParcial IS	2doParcial IS	3erParcial IS	FINAL IS	1erParcial IIS	2doParcial IIS	3erParcial IIS
*								

Esta es la pantalla de actualizar notas, una vez seleccionado el nombre del estudiante se deberá dar "enter" y aparecerán las notas divididas por parcial y por materia recibida. Para ingresar notas en esta pantalla el usuario deberá ubicarse en el cuadro correspondiente al parcial y clase correspondiente, el cuadro viene con una nota predeterminada de cero (0) y el usuario escribirá la nota que corresponde a la materia, posteriormente deberá dar enter para saltar a la línea de abajo y luego cerrar la ventana; los datos se guardaran automáticamente en la base de datos.

**COLEGIO ANGLICANO
PUERTO CABEZAS - RAAN
BOLETIN DE NOTAS**

ID	NOMBRES	GRUPO		APELLIDOS	GRADO	TURNO	PRIMER SEMESTRE				SEGUNDO SEMESTRE			
							IP	IIP	IIIP	FINAL IS	IP	IIP	IIIP	FINAL IIS

Esta es una hoja de reporte de notas de alumnos del centro escolar, esta misma se desplegará cuando ordenemos generar reporte desde el menú principal, al dar click en la opción reportes y luego click sobre la opción boletín.

21/11/2007

**COLEGIO ANGLICANO
PUERTO CABEZAS - RAAN
PLANTILLA DE ESTUDIANTES**

4TO. GRADO

CUARTO A		
0001	YADER	GONZALEZ
<i>Total Grupo</i>		1
Total Grado	1	

5TO. GRADO

QUINTO A		
0008	Ivonne	Siezar
0009	Frank	Sinatra
<i>Total Grupo</i>		2
QUINTO B		
0006	Miguel	Calderon
<i>Total Grupo</i>		1
Total Grado	3	

6TO GRADO

SEXTO A		
0004	Wilber Javier	Bucardo
<i>Total Grupo</i>		1
SEXTO B		
0010	juan	peres
<i>Total Grupo</i>		1
Total Grado	2	

1ER. AÑO

1ER AÑO A		
0005	Alvaro	Ruiz
<i>Total Grupo</i>		1
Total Grado	1	

Este es el reporte de plantilla de estudiantes, donde se refleja a todos los estudiantes matriculados en el colegio este reporte estará clasificado por grupos de clase. En él se visualiza el nombre del alumno, el grupo al que pertenece con su respectiva sección o aula de clases y el Id del estudiante.

Para generar un reporte de plantilla de estudiante se deberán seguir los mismos pasos utilizados para el reporte de notas cuando ordenemos generar reporte desde el menú principal, al dar click en la opción reportes y luego click sobre la opción plantilla.

Catálogo de Materias

Id Materia

Nombre materia

<input type="checkbox"/> 1er.Nivel	<input type="checkbox"/> 5to.Grado
<input type="checkbox"/> 2do.Nivel	<input type="checkbox"/> 6to.Grado
<input type="checkbox"/> 3er.Nivel	<input type="checkbox"/> 1er.Año
<input type="checkbox"/> 1er.Grado	<input type="checkbox"/> 2do.Año
<input type="checkbox"/> 2do.Grado	<input type="checkbox"/> 3er.Año
<input type="checkbox"/> 3er.Grado	<input type="checkbox"/> 4to.Año
<input type="checkbox"/> 4to.Grado	<input type="checkbox"/> 5to.Año

Barra de Botones

Label3

Esta es la pantalla de catalogo de Materias, que se utiliza para crear una nueva materia y asignarla a los grupos donde será impartida. Si se desea asignar una materia a un grupo, el usuario deberá dar click en el botón nuevo de la barra de botones, luego deberá ingresar el Id y el nombre de la nueva materia y tachar los grados en que será impartida dicha materia.

También en esta pantalla se pueden modificar los campos correspondientes a las materias que se imparten por grupo al dar click sobre el botón modificar inmediatamente se activaran todos los campos del formulario y se podrá realizar las modificaciones deseadas, al terminar el proceso como último paso deberá dar click sobre el botón guardar (Este último paso se aplica también para el proceso de crear una nueva materia).

The image shows a software window titled "Catálogo de Grados". Inside the window, there are three input fields: "Id Grado" with a small text box, "Nombre Grado" with a larger text box, and "Nomenclatura" with a small text box. Below these fields is a "Barra de Botones" (button bar) containing several icons: a double left arrow, a single left arrow, a document icon, a printer icon, a close icon (X), a single right arrow, a double right arrow, a refresh icon, and a checkmark icon. To the right of the button bar is a button labeled "Vista Previa".

Este formulario permite ingresar los grados con los que va contar el colegio, para esto el usuario dará click en el botón nuevo, posteriormente deberá digitar el Id del grado así como el nombre de este sin escribir la clasificación del grupo (A, B, C), a continuación escribirá la nomenclatura del grado que tiene que comenzar con una letra N mayúscula seguido de un numero entero, al terminar este proceso se procederá a guardarlo al dar click sobre el botón guardar.

Si el usuario lo desea cuenta con la opción de visualizar los grados existentes dando click en el botón "Vista Previa" que se ubica en la esquina inferior derecha del formulario. Al dar click sobre este botón nos mostrara un pequeño reporte conteniendo los grados guardados actualmente.

Catalogo de Profesores

Id

Nombres

Apellidos

Sexo

Dirección

Telefono

Fec. Nac.

Cedula

INSS

Guia

Barra de Botones

Label11

En esta pantalla el usuario deberá ingresar los datos personales de los profesores así como modificar y eliminar datos guardados anteriormente en la base de datos del colegio.

Para crear, modificar y eliminar un dato se seguirán los mismos pasos utilizados en los formularios anteriores, al dar click sobre el botón del proceso a realizar.

Después de terminar de terminar la acción deseada se deberá finalizar el proceso dando click sobre el botón guardar.

Anexo 3

GLOSARIO

AAN: Análisis del área de negocio se ocupa para identificar en detalle la información y los requisitos de las funciones de área de negocio seleccionadas.

Ámbito del software: Describe la función, el rendimiento, las restricciones, las interfaces y la fiabilidad, se evalúan las funciones del ámbito y en algunos casos se refinan para dar más detalles antes de la estimación.

Análisis y diseño orientado a objetos: La esencia del análisis y diseño orientado a objetos es el considerar tanto el dominio del problema como la solución lógica desde la perspectiva de los objetos (cosas, conceptos o entidades).

Almacenamiento: Término que describe un dispositivo o medio capaz de acceder datos, conservarlos y proporcionarlos cuando se le pidan más adelante.

Análisis de sistema: Investigación detallada, paso por paso, de procedimientos relacionados entre sí para ver qué es lo que se debe hacer y cuál es la mejor manera de hacerlo.

Analista de sistemas: Persona que estudia las actividades, métodos, procedimientos y técnicas de los sistemas de las organizaciones con el fin de averiguar qué acciones se deben emprender y cuál es la mejor forma de llevarlas a cabo.

Archivo: Conjunto de registros relacionados entre sí que tratan como unidad.

Arquitectura: Organización e interconexión de los componentes de los sistemas de computo.

Base de Datos: Conjunto almacenado de las bibliotecas de datos que requieren las organizaciones e individuos para satisfacer sus necesidades de procesamiento y recuperación de información.

Bus: Circuito que constituye un trayecto de comunicaciones entre dos o mas dispositivos.

Byte: Grupo de bits adyacentes, por lo regular ocho, que se manejan como unidad.

Casos de Uso: Es una operación o tarea especifica que se realiza tras una orden de algún agente externo, sea desde una petición, de un actor o bien desde la invocación desde otro caso de uso.

Clase: Una clase de un objeto es una atracción que describe un grupo de instancias, con propiedad (atributos) comunes, comportamiento (operaciones) común, relaciones comunes con otros objetos y una semántica común. la diferencia entre una clase y una instancia esta en el grado de abstracción.

Código: Conjunto de reglas que delinear la forma en que se pueden representar los datos; también, reglas que se emplean para convertir los datos de una representación a otra.

Clientes: Interactúan con el usuario, usualmente en forma grafica. Frecuentemente se comunican con procesos auxiliares que se encargan de establecer conexión con el servidor.

Case (compute-aided software engineering): Ambiente integrado de desarrollo de software que permite la codificación de programas a partir de lenguajes gráficos de modelación.

Componentes de software: Los componentes representan todos los tipos de elementos de software que entran en la fabricación de aplicaciones informáticas, pueden ser simples archivos, paquetes de Ada, bibliotecas cargadas dinámicamente.

Contador: Dispositivo que se usa para representar el número de ocurrencias de un hecho.

Clase o clase de objeto: Es una abstracción que describe un grupo de instancias con propiedades (atributos) comunes, comportamiento común, relaciones comunes con otros objetos una semántica común.

DRA: Desarrollo rápido de aplicaciones, es un modelo de proceso del desarrollo del software lineal secuencial que enfatiza un ciclo de desarrollo extremadamente corto de 60 a 90 días.

Dependencia o instancia: Es una forma muy particular de relación entre clases, en la otra, es decir, se instancia (se crea) se denota con una flecha punteada.

Diseño conceptual: Es el enfoque que se realiza desde la perspectiva del usuario y/o del negocio. El objeto principal consiste en la definición del proyecto y de los conceptos de la solución escenarios y modelos son usadas para suavizar este entendimiento de manera que cada una de las entidades involucradas sepan que es lo que se necesita de la solución.

Diseño lógico: Es el proceso de tomar los requerimientos de usuarios obtenidos en el diseño conceptual y mapearlos a sus respectivos objetos de negocios y servicios. Los objetos y servicios, la interna de usuarios y la base de datos lógico con el conjunto de elementos identificados y diseñados en esta perspectiva.

Diseño físico: es el proceso en donde los requerimientos del diseño conceptual son puestos en una forma intangible. Este define como los componentes de la solución, así como la interfaz de usuario y la base de datos física trabajan juntos. Desempeño, implementación, ancho de banda, estabilidad, adaptabilidad y mantenimiento son todos resueltos e implementados a través de diseño físico.

Diseño de sistema: Creación de soluciones opcionales a los problemas que descubre el análisis de sistema. La recomendación final de diseño se basa en la efectividad con respecto al costo y en otros factores.

Documentación: Durante el análisis de sistema y programación subsecuente, preparación de los documentos que describen el sistema, los programas preparados los cambios que se efectúan más adelante.

Extend: se recomienda utilizar cuando un caso de uso es similar a otro.

Gantt: Gráfica de tiempo para cada función, para cada individuo que trabaje en el proyecto o para todo proyecto.

Gestor de Base de Datos: Lugar donde se almacena la base de datos en este caso la base de datos estará almacenada en el servidor y será manipulada por el usuario.

Hardware: Equipo físico como por ejemplo dispositivos electrónicos, magnéticos y mecánicos.

Interfaz: Frontera compartida (frontera entre dos sistemas o dispositivos).

Intranet: Red local que utiliza los mismos protocolos de internet, independientes de que esté o no conectada a internet.

Métrica del proyecto: Medida cuantitativa del grado en que un sistema, un componente o un proceso posee un atributo dado. Las métricas y los indicadores derivados de ellos los utilizan un gestor de proyectos y un equipo de software para adaptar el flujo de trabajo del proyecto y las actividades técnicas.

Modelo conceptual: Es una representación de conceptos y objetos en el dominio del problema, como libro y biblioteca.

Modelo de la empresa: Este se define en la clásica jerarquía de unidad de negocio (por ejemplo: un organigrama), donde cada caja del organigrama representa un área del negocio de la empresa.

Modelado de datos: Flujo de información definido como parte de la fase de modelado de gestión se refina como un conjunto de objetos de datos necesarios para apoyar la empresa se definen las características (atributos) de cada uno de los objetos relacionados entre esos objetos.

Proceso de archivos: Actualización de archivos maestros sobre datos para reflejar los efectos de las transacciones actuales.

Proceso de datos: Una o más operaciones realizadas sobre datos para alcanzar un objetivo deseado.

Programa: Conjunto de instrucciones en secuencia que hacen que la computadora lleve a cabo determinadas operaciones.

Programar: Diseñar, escribir y probar una o más rutinas.

PEI: Planificación estratégica de la información, crea un modelo de datos, o nivel del negocio que define los objetos de datos clave y sus relaciones entre ellos y con otras áreas del negocio. Los términos "objetivos" y "metas" toman un significado específico en la PEI.

Red: Interconexión de sistemas de computo o dispositivos periféricos en localidades dispersas que intercambian datos cuando es necesario para llevar a cabo las funciones de la red.

Registro: Conjunto de elementos de información relacionados entre si que se tratan como unidad. Dispositivos capaces de almacenar una cantidad específica de datos.

Sistema: Agrupación de métodos y procesamientos integrados para formar un ente organizado. Agrupación organizada de personas, métodos, maquinas y materiales reunidos para lograr un conjunto de objetivos específicos. Es un conjunto de componentes que interactúan entre sí para lograr un objetivo común.

Servidor: Proporciona un servicio al cliente y devuelven los resultados. En algunos casos existen procesos auxiliares que se encargan de recibir las solicitudes del cliente, verificar la protección, activar un proceso servidor para satisfacer el pedido, recibir su respuesta y enviarla al cliente.

Usuario: Persona que interactúan con el software.

UML: Es una notación para la descripción e interpretación visual de las partes relacionales y acciones que conforman una aplicación de software. Es un lenguaje flexible de modelado que permite definir modelos arbitrarios.

Ventana: Parte de la pantalla de exhibición visual que se emplea para mostrar la situación actual de una aplicación que interesa al usuario. La pantalla se puede dividir en varias ventanas para mostrar simultáneamente la situación de las distintas aplicaciones en un paquete integrado de software.

Viabilidad: Estudio minucioso de la parte técnica legal, operativa, económica de un producto basado en la computadora si está plagado. La viabilidad y el análisis de riesgo están relacionados de muchas maneras. Si el riesgo del proyecto es alto, la viabilidad de producir software de calidad se reduce.

Anexo 4
Contrato de Trabajo

Nosotros, James Edwards Mccoy, mayor de edad, viudo, padre de la Iglesia Anglicana, de este domicilio que se identifica con cédula del consejo supremo electoral N° 601-240935-0001Y, actuando en nombre de la Escuela Episcopal Santísima Trinidad Anglicana en carácter de supervisor general y que en lo sucesivo de este contrato de trabajo se denominará la escuela, y

_____ Mayores de edad _____

De este domicilio, que se identifican con números de cédula del consejo supremo electoral N° _____ actuando en su propio nombre y representación y que en lo sucesivo se denominará el contrato hemos acordado el siguiente contrato de trabajo el cual se regirá por el código del trabajo y por las cláusulas siguientes.

Cláusula Primera

Objetivo del contrato. La escuela contrata a los desarrolladores para que presten sus servicios profesionales con el cargo de _____ en el colegio Episcopal Santísima Trinidad Anglicana de esta ciudad.

Cláusula segunda

Duración del contrato. El presente contrato de trabajo será por tiempo determinado y tendrá una duración efectiva de 6 meses a partir de la firma del presente contrato, hasta la fecha del 26 de noviembre 2007.

Cláusula tercera

Del pago y forma de pago. El colegio pagará a los desarrolladores en concepto de pago total la cantidad de \$ _____

Córdobas (C\$ _____) el día 26 de noviembre del año 2007 y a retirarse en la dirección administrativa del colegio.

Cláusula cuarta

De las prestaciones sociales. El colegio, al finalizar el termino de este contrato de trabajo estará solvente de todo pago por indemnización respectiva a lo que este tenga derecho, de conformidad a las disposiciones establecidas en el código de trabajo.

Cláusula quinta

Rescisión del contrato. Las partes podrán rescindir del presente contrato por mutuo acuerdo o por solicitud de una de las partes con quince días de antelación.

Cláusula Sexta

Aceptación. Las partes aceptan todas las cláusulas en general, y cada una en particular, establecidas en el presente contrato.

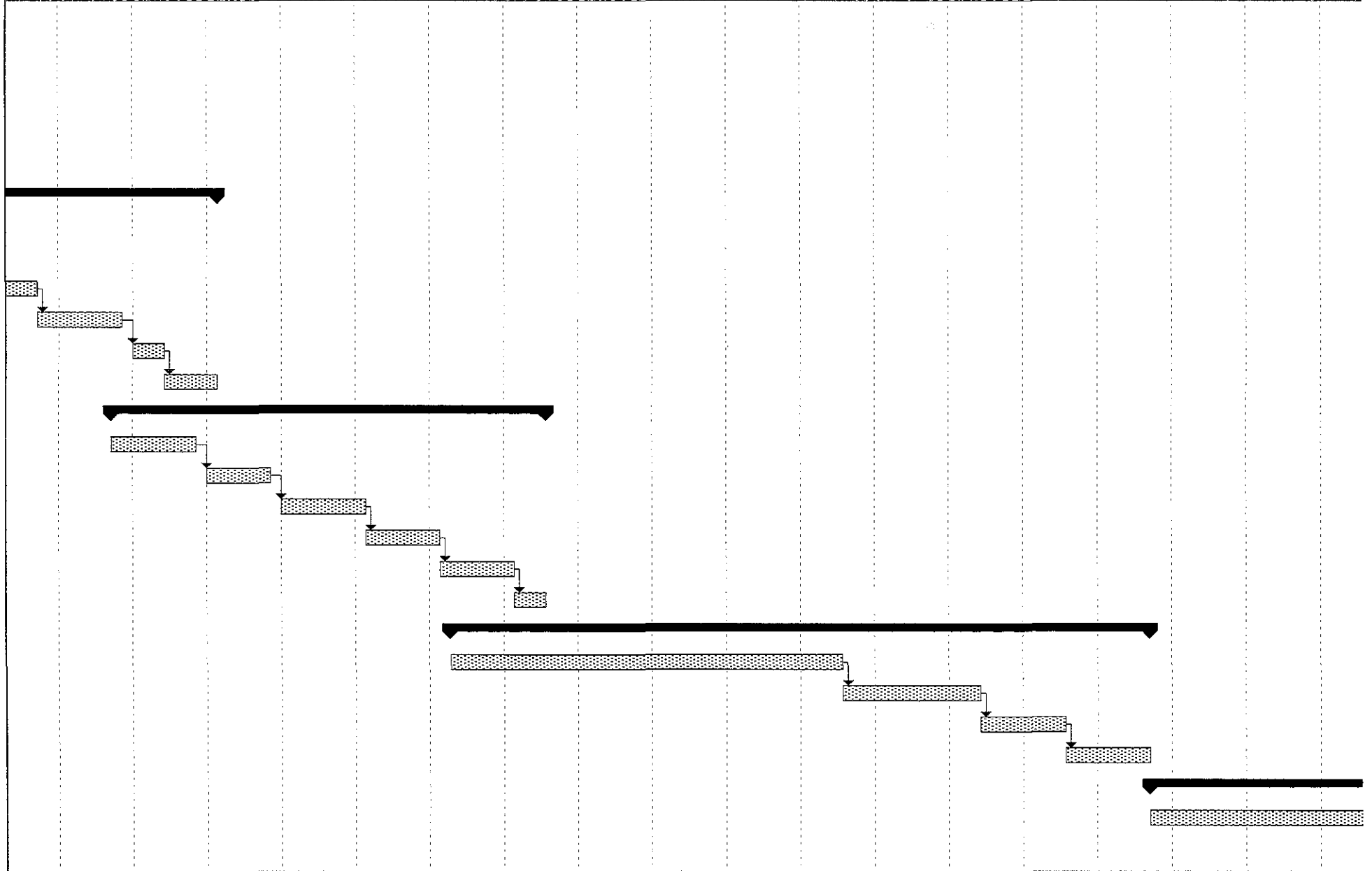
En fe de lo anterior y leído que fue el presente contrato, lo encontramos y firmamos todos, en tres originales y en idioma español, en la ciudad de Bilwi, Puerto Cabezas a los _____ días del mes de _____ del año 200_____.

**Revdo. Padre James Edwards M
Supervisor General del colegio
Epis.St.Trinidad Anglicana
Pto. Cabezas**











Desarrolladores

Cc: Mitrab
Cc: Archivo

1 '07 16 jul '07 23 jul '07 30 jul '07 06 ago '07 13 ago '07 20 ago '07 27 ago '07 03 sep '07 10 sep '07 17 sep '07 24 sep '07 01 oct '07 08 oct '07 15 oct '07 22 oct '07 29 oct '07 05 nov '07 12 nov '07



Proyecto: Colegio Anglicano
 Fecha: jue 22/11/07

Tarea		Progreso		Resumen		Tareas externas		Fecha límite	
División		Hito		Resumen del proyecto		Hito externo			



DATOS DEL CLIENTE

Señores : COLEGIO ANGLICANO
 Atención : DIRECTOR

DATOS DE LA PROFORMA


Nº. Ref. : AV-JL291007
 Fecha : 29 de Octubre de 2007
 Asunto : Presentación de oferta

COMTECH tiene el gusto de ofertarle, los accesorios y equipos de computación con las especificaciones técnicas solicitadas a precios económicos.

COTIZACIÓN N°. JL2911007

Cant.	Descripción	V. Unitario \$	V. Total \$
01	COMPUTADOR Tarjeta Madre Upgradable Procesador INTEL CELERON 2.8GHZ Memoria cache de 256KB 533Mhz front side Bus Memoria 512MB DDR KINGSTON exp 2.0GB 160GB Disco Duro SATA 7200rpm Case ATX Minitorre Unidad Óptica: MULTI DVD+/-RW(DL), SAMSUNG Tarjeta de RED 10/100 VIA 6103 dual speed full duplex Tarjeta de video SDRAM 32MB compartido Sistema de video: Desempeñado en 2D y 3D Sonido AC'97 sound codec soporta 2 canales 1 paralelo, 1 serial, 2 PS/2 para mouse y teclado 6 puertos USB Arquitectura PCI: 3 slot PCI, 1 SLOT PCI-EX Teclado, Mouse con Scroll, Almohadilla, Parlantes MONITOR SVGA CRT 15" GRATIS MEMORIA USB 1GB, KINGSTON <u>NO INCLUYE FLOPPY DRIVE</u> ESTOS EQUIPOS NO INCLUYEN SOFTWARE, LA EMPRESA NO SE HACE RESPONSABLE POR LA INSTALACION DE SOFTWARE NO ORIGINAL EN ELLOS.	369.00	369.00
02	MISMA CONFIGURACION SOLAMENTE CON CD-ROM 52X SAMSUNG	354.00	708.00
03	UPS CON ESTABILIZADOR INTEGRADO 505VA CDP 5 TOMAS RESPALDO DE 5 - 20 MINS	38.00	114.00

COMTECH, S.A.
 Jessy López Santana
 Asesora de Ventas
 Tel: 2525767 Fax: 2706224

01	<p>IMPRESORA HP 1018 LASERJET /CB419A#ABA MARCA: HEWLETT PACKARD MODELO: HP LaserJet 1018 - printer - B/W - laser Printer Type: Personal printer - laser - monochrome Form Factor: Desktop Dimensions (WxDxH): 14.6 in x 9.5 in x 8.2 in Weight: 11 lbs Localization: English / United States Max Media Size (Standard): Legal Max Media Size (Custom): 8.5 in (Legal) x 14 in (Legal) Print Speed: Up to 12 ppm - B/W - A4 (8.25 in x 11.7 in) Max Resolution (B&W): 600 dpi x 600 dpi Image Enhancement Technology: HP FastRes 1200 Interface: Hi-Speed USB Processor: 234 MHz RAM Installed (Max): 2 MB (2 MB) Media Type: Envelopes, transparencies, plain paper, business cards, cards Total Media Capacity: 150 sheets Monthly Duty Cycle: 3000 pages System Requirements: Microsoft Windows 98 Second Edition, Microsoft Windows 2000, Microsoft Windows Millennium Edition, Microsoft Windows XP Manufacturer Warranty: 1 year warranty</p> 	94.00	94.00
01	TONER HP Q2612A (2,500PGS)	72.00	72.00
01	SWITCH 3COM 8PUERTOS	60.00	60.00
01	CABLE UTP METRO	0.40	0.40
01	CONECTOR RJ45	0.25	0.25

COMTECH S.A.
 Jessy López Santana
 Asesor de Ventas
 Tel: 2525767 Fax: 2706224

▶▶ **ESTOS PRECIOS MÁS EL 15% DE I. V. A**

- ▶▶ **Tiempo de Entrega: 24 HORAS DESPUES DE SU APROBACION**
- ▶▶ **Forma de pago: Contado y/o ck a/n de COMTECH**
- ▶▶ **GARANTIA DE 1 AÑO EN CPU Y MONITOR, 1 MES PARA LA FUENTE DE PODER, 3 MESES PARA TECLADO, MOUSE Y PARLANTES**
- ▶▶ **Precios cotizados en dólares / Tipo de cambio paralelo 18.95**
- ▶▶ **Esta oferta es válida por 15 días**
- ▶▶ **Para efectos de cubrir la garantía se le recomienda instalar el equipo en un toma corriente polarizado, protegerlo con un estabilizador y realizarle mantenimiento correctivo de 3 a 6 meses.**



Me despido


JESSY LOPEZ SANTANA
Asesora de Ventas
Tecnología Computarizada-COMTECH
PBX 2674012 Ext. 110

COMTECH, S.A.
Jessy López Santana
Asesora de Ventas
Tel: 2525787 Fax: 2706224



PRINCIPAL ALTAMIRA D,ESTE No. 589 Ferrreteria Sinsa 25vrs. arriba
 Tels.:PBX (505) 267-4012 Fax: (505) 270-6224 E-Mail : COMTECH@CABLENET.COM.NI
 RUC No. 260201-9015 Apartado No. -

ente : **-COLEGIO ANGLICANO**
 ección :
 nción a:
 éfono/Fax:

PROFORMA No. 00-0000043222
 Fecha : 29 de Octubre de 2007
 Ejecutivo de Venta: JESSY LOPEZ
 Condición de pago CONTADO
 Autorizado Por: _____

CODIGO	DESCRIPCION	UBIC.	CANTIDAD	PRECIO	TOTAL
05001-117	MICROSOFT OFFICE HOME AND STUDENT 2007 - OEM - SPANISH -		3	159.00	477.00
05001-037	SOFTWARE NORTON ANTIVURUS 2007 /10725856-SL		3	49.00	147.00
05001-003	SOFTWARE WINDOWS XP PRO SPANISH OEM E85-04781		3	164.00	492.00

ODUCTOS NO GRAVABLES

: TC18.95

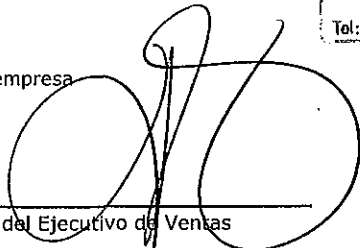
COMTECH, S.A.
 Jessy López Santana
 Asesora de Ventas
 Tel: 2525767 Fax: 2706224

Sub - Total	1,116.00
Descuento	0.00
IGV	167.40
TOTAL PROFORMA US\$	1,283.40

lida solamente con el sello de la empresa

29/10/2007 11:04

Page 1 of 1


 Firma del Ejecutivo de Ventas

 Firma del Cliente

El valor de esta cotización es válida por ocho días

COTIZACION No.: C00666

07/11/2007

Nombre del Cliente: Colegio Janglican

Sucursal

Persona que solicito el Servicio:

Proyecto:

Telefono:

Fax:

Celular:

Item	Código	Descripción	Cantidad	Precio	Total de Linea
1.00		icmp8p8srdPlug,oval entry solid 8p8c	1.00	0.44	0.44
2.00		icrw11srwh Raceway,3/4"wx1/2"hx6"1 white 20pcs	1.00	4.79	4.79
3.00		icrw12srwh Raceway,11/4"wx3/4"hx6"1white 20 pcs	1.00	7.12	7.12
4.00		icrw13srwh Raceway,13/4x11/4"white 20 pcs	1.00	10.53	10.53
5.00		icrw11eowhFlat,elbow,raceway,3/4" white 10pcs	1.00	1.17	1.17
6.00		icrw11icwh Cover.inside, corner,raceway,3/4" white	1.00	1.20	1.20
7.00		icrw11ocwhOutside,corner,cvr,raceway,3/4"white	1.00	1.20	1.20
8.00		icrw12eowh Flat,elbow.raceway,11/4"white 10 pcs	1.00	1.29	1.29
9.00		icrw12icwh cover, inside corner, raceway 1 1/4	1.00	1.29	1.29
10.00		icrw12ocwhOutside,corner,cvr,raceway,11/4"white	1.00	1.29	1.29
11.00		icrw13eowhFlat,elbow,raceway,13/4" white 10pcs	1.00	1.39	1.39
12.00		icrw13icwh Cover,inside,corner,raceway,13/4"white	1.00	1.39	1.39
13.00		icrw13ocwhOutside,corner,cvr,raceway,13/4"white	1.00	1.39	1.39
14.00		cm-00424p:Cable utp cat 5e blue panduit	1.00	0.26	0.26

Sub Total: 34.75

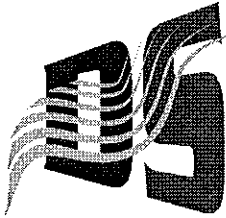
Descuento: 0.00

15% de IVA: 5.21

Total Cotizacion: 39.96

TREINTA Y NUEVE Dólares 96/100 M.N.

Yansis Martinez
Asesora de Ventas
Tel.: 278-3039 ext 9110



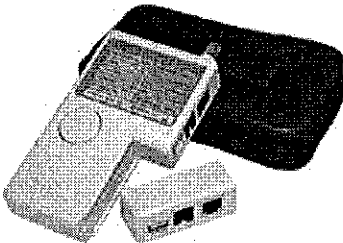
DATASYSTEM

Managua, 15 de noviembre del 2007

COLEGIO ANGLICANO

Atención:

Telefax: 7922469

Cant	Descripción del Equipo	Valor Unitario	Total US\$
01	Remote Multicaule Tester <ul style="list-style-type: none">• The remote cable tester checks the cable before and after installation.• 14 LED's confirm testing status and shut off automatically to maximize power saving.• Testing...RJ45 = USOC 4/6/8, 10Base-T, EIA/TIA 568 A/B, Token Ring, RJn, BNC, USB• LED's = Testing, short, connection, non-parallel 25/50 Ohm termination, screening each conductor. 	58.00	58.00
01	CORTADORA AW250NXT02	10.00	10.00
		Subtota	68.00
		Iva	10.20
		Total	78.20

GENERALIDADES:

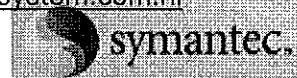
- Precios en dólares. Pagaderos en Córdobas al cambio Paralelo a Bancentro.
- Validez de la Oferta: 15 días a partir de la fecha de recepción.
- Garantía: 3 Meses
- Forma de Pago: Contado. Efectivo y/o Cheque a nombre de DATASYSTEM, S.A

En espera de poder servirle, le saluda.

Francis Sanchez
Cuentas corporativas

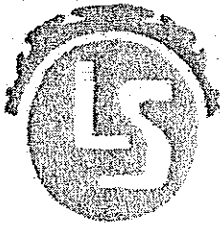
Cel: 6266629 Telefax: 270-6039

fsanchez@datasystem.com.ni Web site: www.datasystem.com.ni



ALTAMIRA D'ESTE DONDE FUE LA DISTRIBUIDORA VICKY. PBX: (505) 270.60.39

Web: www.datasystem.com.ni Email: ventas@datasystem.com.ni



LIBRERÍA SAMUEL & CIA. LTDA

SUCURSAL ALTAMIRA

RUC 200693 - 9514

Costado Este del BANPRO casa 518.

Telf. 2703992 - 2709679, Managua, Nicaragua.

Factura Proforma

000029958

CLIENTE : COLEGIO ANGLICANO

VENCE 29/11/2007 Zona 1

Teléfono :

Celular :

FECHA 14 11 2007

1	MER-3619	GUILLOTINA MERLETTO 3619 DE 12 PULG.	329.29	329.29
10	CLAS-TC	PAPEL BOND 40 T/C CLASSIC 96% BLANC. 8 1/2X11	73.84	738.40



Observaciones:

Esta Proforma tiene validez por 15 días

ELABORADO POR

Valeria J.

C\$ por Dólar
* Productos no Gravables

SUB-TOTAL C\$	1,067.69
I.V.A C\$	160.15
TOTAL C\$	1,227.84

Anexo 5

Bibliografía

www.monografias.com

www.lawebdelprogramador.com

www.ilustrado.com

Modulo:

Ingeniería Web 2007, Autor: Ing. Fausto Quiñonez.

Aplicaciones Web 2007, Autor: Ing. Fausto Quiñonez.

Intranet 2007, Autor: Msc. Irineo Moody.

**Evaluación de Proyectos Informáticos 2007, Autor: Ing. Manuel
Rojas.**

UML 24 Horas