

Universidad Nacional Autónoma de Nicaragua (UNAN-León)

Facultad de Ciencias Puras
Departamento de Computación



Tema:

“Sistema de Automatización para la Elaboración e Impresión de Recibos y Matrículas del Instituto Socorro Santana Solís (ISSS)”.

Monografía para optar al Título de Ingeniero en Sistemas de Información.

Integrantes:

Br. María del Carmen Morales Maradiaga.

Br. Silvia Vanessa Morales Osejo.

Tutor:

MSc. Ernesto Espinoza Montenegro.

León, Nicaragua 2007



INDICE DE CONTENIDO

Contenido	Páginas
II. Introducción	1
III. Antecedentes	2
IV. Justificación	3
V. Objetivos	4
a. General	4
b. Específico	4
VI. Marco Teórico	5
a. Material y Método	12
b. Metodología de trabajo	13
c. Estructura del modelo de vida clásico	15
VII. Análisis del sistema	17
a. Descripción del Sistema	17
b. Diccionario de Datos	21
c. Diagrama Entidad – Relación	22
d. Especificación de Requisitos Software	23
VIII. Conclusiones	51
IX. Recomendaciones	52
X. Bibliografía	53
XI. Anexos	54
a. Diseño relacional	48
b. Diseño Procedimental	49
c. Diseño Arquitectónico	50
d. Codificación	65



INDICE DE TABLAS

Contenido	Páginas
I. Diseño	45
a. Diseño de Datos	45
i. Tabla Estudiante	45
ii. Tabla Matriculas	46
iii. Tabla Aulas	46
iv. Tabla Recibos	47
v. Tabla DetalleRecibos	47
vi. Tabla Aranceles	47



INDICE DE INTERFAZ

Contenido	Páginas
I. Anexos	54
a. Diseño de Interfaz	54
i. Formulario Principal	54
ii. Formulario Clave	54
iii. Formulario Presentación	55
iv. Formulario Nuevo Recibo	55
v. Formulario Anular Recibo	56
vi. Formulario Catalogo de Aranceles	56
vii. Formulario Nuevo Arancel	57
viii. Formulario Modificar Aranceles	57
ix. Formulario Nueva Matricula	58
x. Formulario Matriculas X Fecha	58
xi. Formulario Modificar Matriculas	59
xii. Formulario Catalogo de Estudiante	59
xiii. Formulario Nuevo Estudiantes	60
xiv. Formulario Modificar Estudiantes	60
xv. Formulario Catalogo de Aulas	61
xvi. Formulario de Aulas	61
xvii. Formulario Modificar Aulas	62
xviii. Formulario Recibos X Fecha	62
xix. Formulario Mostrar Recibo	63
xx. Formulario Estado de Cuenta	63
xxi. Formulario Reporte Estado de Cuenta	64
xxii. Formulario CambiarUsuario	64



Dedicatoria.

- A Dios, por estar conmigo en cada instante y darme la fuerza y la fe para seguir adelante.

*- A mis padres **Victor Morales** y **María Jesús Maradiaga**, quienes con su entrega y sacrificio hicieron posible que alcanzara mi primer meta. Gracias a ellos he aprendido que todo lo bueno que la vida te puede dar cuesta, pero que con esfuerzo, tolerancia y perseverancia todo se puede lograr.*

*- A todos mis amigos, pero en especial mención al **Lic. Carlos Gómez** por el apoyo que siempre nos brindó.*

María del Carmen Morales Maradiaga



Dedicatoria.

*A mis padres: **Pablo Morales** y **Silvia Osejo** por haberme inculcado valores morales, proporcionarme la educación que hoy tengo, por apoyar mis decisiones y ser quien con amor, cariño y sacrificio logro financiar mis estudios universitarios.*

*A **Dios** que me ha dado sabiduría para distinguir entre el bien y el mal, fuerzas para seguir adelante en los momentos difíciles de la carrera.*

*A mis Amigos por estar conmigo en todos los momentos especialmente al **Lic. Carlos Gómez** por apoyarnos siempre.*

Silvia Vanessa Morales Osejo



Agradecimiento.

A las personas que nos han ayudado para la formulación y desarrollo de nuestro proyecto. A nuestros amigos, compañeros y profesores por medio de este documento.

*Al MSc. **Ernesto Espinoza Monterrey** por habernos guiado en la realización de nuestro trabajo monográfico y lograr culminar nuestra carrera.*

*También nos gustaría mencionar al MSc. **Francisco Danilo Padilla** por el tiempo dedicado a la revisión y corrección de este documento tanto en las horas de clases como en sus momentos libres.*



I. INTRODUCCIÓN

El Instituto Socorro Santana Solís (ISSS) fue fundado en 1974, primeramente con el nombre de Luis Manuel Debayle y es uno de los centros educativos que se ha destacado en los diferentes concursos orientados por el Ministerio de Educación (MECD).

Actualmente esta ubicado en Telica y desde sus primeros días de funcionamiento todas las actividades que se desarrollan se han llevado a cabo de manera manual, es por ello que se hace necesario emplear el uso de tecnología reciente, puesto que se dispone de estos recursos se pretende crear una aplicación que permita al estudiante realizar cualquier gestión de una manera fácil y rápida.

En la parte de matrículas el centro está interesado en poder administrar información relacionada con los datos personales de los estudiantes actualmente matriculados y los datos propios de cada una de las matrículas.

En la parte de caja, el centro está interesado en controlar los pagos que los estudiantes están obligados a efectuar a lo largo del año escolar, por ejemplo: Pago de la matrícula, Pago papelería, Pago de colegiaturas, Otros pagos...

En el Instituto se ofrecen matrículas de secundaria (matutino-vespertino) cuenta con una población estudiantil de aproximadamente 1025 estudiantes, en el laboran 31 personas entre docentes, personal administrativo y de apoyo.

Esta aplicación se desarrollo con el fin de mejorar los tiempos de respuesta ante una demanda cada vez más creciente y exigente, proporcionando un servicio fiable y de mejor calidad.



II. ANTECEDENTES

El mundo atraviesa una etapa de profundos cambios, en un período como en el que actualmente toca vivir, caracterizado entre otras cosas por mercados muy competitivos, todas las actividades deben elevar sus niveles de calidad si se desea incorporar a la dinámica de la economía global. La Educación no es una excepción y es por ello que el Instituto Socorro Santana Solís se percató de la necesidad de analizar y evaluar los procedimientos que lleven a mejorar la calidad del servicio que ofrecen a sus usuarios, con la automatización del área de matrículas y de caja se logrará disminuir sustancialmente el tiempo que requiere el proceso de matrículas, y lo que es más importante llevar un mejor control sobre los pagos de aranceles que los estudiantes hacen a lo largo del año escolar, así como también la recuperación o búsqueda de información, y la emisión de reportes como por ejemplo el estado de cuentas por estudiantes.

La forma tradicional que ha utilizado el Instituto para matricular y elaborar recibos requiere de grandes inversiones de tiempo, recursos económicos y esfuerzo humano lo cual evita que el centro logre moverse con mayor facilidad en esas áreas en particular.



III. JUSTIFICACIÓN

Con el desarrollo de esta aplicación se pretende disminuir el tiempo del proceso de matricula de cada estudiante, elaborar e imprimir un determinado recibo, así como tener un mejor control sobre el número de estudiantes matriculados actualmente en el centro y sobre todo controlar de manera eficiente los pagos de éstos al centro escolar.

De esta manera el Instituto logrará alcanzar un nivel de atención mucho más rápido y eficiente, cumpliendo así con sus objetivos prioritarios de dar su aporte al desarrollo de la ciencia y brindar una mejor atención a los estudiantes y a los padres de familia.



IV. OBJETIVOS

Objetivo general:

- Crear una Aplicación que permita la Elaboración e Impresión de Recibos y Matrículas Escolares.

Objetivos específicos:

- Diseñar un sistema capaz de llevar un registro de todos los pagos que realizan los estudiantes del centro.
- Desarrollar una aplicación que almacene y maneje los datos personales de cada estudiante.
- Diseñar una aplicación que permita matricular a un estudiante.



V. MARCO TEÓRICO

La plataforma .NET Framework está diseñada para simplificar el desarrollo de aplicaciones en el entorno distribuido y consta de un motor de ejecución de código de alto rendimiento y de clases Framework con Código reutilizable, una de las principales características de esta plataforma es la facilidad de uso y los lenguajes utilizados para acceder a los datos, el acceso a datos es la esencia de cualquier aplicación del mundo real, sin duda alguna siempre que hagamos algo en .NET Framework será mediante las clases que este "marco" nos proporciona, las cuales están contenidas en un espacio de nombres (namespace), que nos permite hacer uso de ellas de una manera organizada.

Para proporcionar el acceso a datos, los servicios del entorno incluyen una biblioteca de clases ADO.NET, el cual es el nuevo estándar de Microsoft para trabajar con Bases de Datos y se basa en una nueva tecnología de acceso a datos de Microsoft llamada ADO+.

ADO.NET representa el sustrato que compondrá la base de las aplicaciones .NET compatibles con datos. A diferencia de ADO, ADO.NET se ha diseñado siguiendo específicamente unas directrices más generales y menos orientadas a la base de datos.

¿Por qué utilizar ADO.NET?

Una de las razones principales es porque reúne todas las clases que permiten el manejo de datos. Estas clases representan los objetos que contienen datos y muestran las capacidades normales de las bases de datos: índices, ordenación, vistas, y la otra razón es porque permite enviar las instrucciones SQL a la base de datos mediante los objetos que este nos proporciona, de esta manera, las clases de acceso a datos están todas contenidas en el espacio de nombres System.Data, en este espacio a su vez



existen otros espacios de nombres que contienen clases especializadas según el proveedor de datos que queramos usar, ejemplo si queremos tener acceso a datos de base SQL Server utilizaremos el espacio de nombres System.Data.SqlClient, estos espacios de nombres especializados nos permiten conectar con la base de datos, traer los datos indicados a la memoria y después de manipularlos, guardarlos en la base de datos, si es que realmente hemos hecho algún cambio, inserción o eliminación.

ADO.NET utiliza algunos objetos ADO, como Connection y Command, y también agrega objetos nuevos. Algunos de los nuevos objetos clave de ADO.NET son DataSet, DataReader y DataAdapter.

- El objeto Connection representa una conexión física a una base de datos, lo habitual es que el objeto Connection lo instanciamos usando la cadena de conexión adecuada, en esa cadena de conexión indicaremos (como mínimo) el origen de datos y el usuario o password que debemos usar para acceder a la base de datos.
- El objeto Command representa una directiva para recuperar (select) los datos que queremos obtener de la base de datos o los necesarios para actualizar (comando update) eliminar (comando delete) e insertar nuevos datos (comando insert).
- El objeto DataSet es una entidad independiente y se puede considerar como un conjunto de registros que siempre está desconectado y que no sabe nada sobre el origen y el destino de los datos que contiene. Dentro de un objeto DataSet, de la misma manera que dentro de una base de datos, hay tablas, columnas, relaciones, restricciones, vistas, etc.



- El objeto DataAdapter es el que realmente realiza la conexión, obtención de datos para llenar el objeto DataSet, posteriormente, se vuelve a conectar a la base de datos para actualizar los datos a partir de las operaciones realizadas en los registros contenidos en el objeto DataSet.
- Objetos DataReader. Proporcionan una forma de leer una secuencia de registros de datos sólo hacia delante desde un origen de datos SQL Server

Objeto	Descripción
Connection	Establece una conexión a un origen de datos.
Command	Ejecuta un comando en un origen de datos
DataReader	Lee datos de forma forward-only y read-only.
DataAdapter	Llena un DataSet y realiza las actualizaciones necesarias en la base de datos

En definitiva ADO.NET es una solución bastante distinta de ADO, debido a que posee un nuevo modelo de programación para el acceso a datos. No obstante, una vez que hayamos comenzado a utilizar ADO.NET, observaremos que cualquier tarea de ADO resulta de extrema ayuda para generar aplicaciones eficaces y para resolver antiguos problemas de una forma distinta, aunque de manera más elegante y consistente.

También hacemos uso del compilador Microsoft Visual Basic .Net, el cual es un lenguaje orientado a objetos y eventos que soporta encapsulación, herencia y polimorfismo, siendo una mejora de Visual Basic y formando parte de Microsoft Visual Studio.NET, compartiendo el entorno de desarrollo con Microsoft Visual C++ .NET, Microsoft Visual C# .NET, etc.



Visual Basic .NET contiene dos archivos:

- Un archivo de proyecto .vbproj, el cual contiene información específica para una determinada tarea de programación.
- Un archivo de solución .sln, el cual contiene información relacionada con uno o más proyectos. Este tipo de archivo puede administrar varios proyectos relacionados entre sí y son similares a los archivos de grupos de proyecto (.vbg) en Visual Basic 0.6.

Como gestor de base de datos utilizamos Microsoft SQL Server 2000 Es un sistema de gestión de bases de datos (SGBDR), desarrollado por Microsoft, que permite, como su propio nombre indica, la gestión de un entorno de bases de datos relacional. Este abarca, tanto el área de diseño, como la de administración, proporcionando un interfaz bastante amigable con el usuario, utiliza el lenguaje SQL para la definición y manejo de los datos, dispone de una parte servidora (Server) que se encarga de atender a los procesos clientes, que son los que realizan las peticiones a éste; es decir; sigue una arquitectura Cliente / Servidor.

Para acceder a SQL Server se dispone de dos modos de autenticación: Autenticación con Windows de Microsoft y Autenticación en modo mixto. En la Autenticación con Windows, el sistema operativo es el responsable de identificar al usuario, SQL Server comprueba que el usuario fue validado por Windows NT/2000 y permite el acceso basado en esa identificación, integrando su proceso de seguridad en el inicio de sesión con el de Windows para proporcionar estos servicios. Los atributos de seguridad en la red se validan por medio de un sofisticado proceso de cifrado proporcionado por Windows NT/2000.

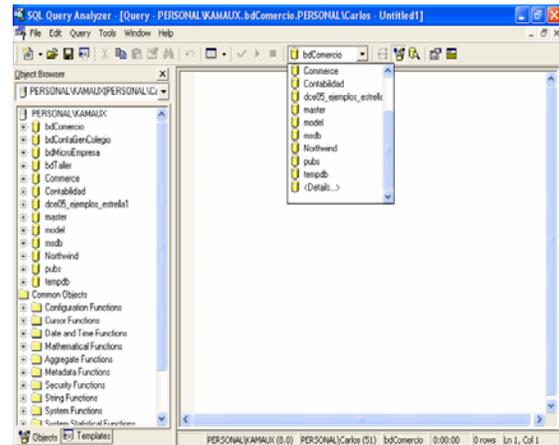


Con la Autenticación en modo mixto los usuarios pueden acceder a SQL Server utilizando la Autenticación de Windows o la Autenticación de SQL Server. Al usar la Autenticación en modo mixto, si se produce una conexión desde un sistema inseguro, SQL Server identifica el inicio de sesión comprobando si dicha cuenta se ha configurado con el acceso solicitado por el usuario. SQL Server realiza la identificación de la cuenta comparando el nombre del código y contraseña que el usuario proporciona al conectarse a SQL Server con la información de la cuenta almacenada en la base de datos. Si la cuenta para el inicio de sesión no ha sido configurada por el usuario o si el usuario no proporciona el nombre correcto y su contraseña, se rechaza dicho acceso.

Algunos desarrolladores de aplicaciones prefieren usar la seguridad de SQL Server para sus aplicaciones porque simplifica la seguridad de las mismas. Cuando las aplicaciones utilizan la seguridad de SQL Server (dentro de una red de confianza), los desarrolladores de aplicaciones no tienen que proporcionar autenticación de seguridad dentro de la propia aplicación, lo que simplifica el trabajo.

En el área administrativa se dispone del analizador de consultas: (query analyser) el cual es una herramienta que permite editar y ejecutar sentencias Transact SQL.

Podemos observar en ella un editor, que sirve para ejecutar las sentencias, una barra de menú, con las opciones disponibles, y una barra de herramientas con íconos de acceso rápido a ciertas utilidades.



Las **vistas** son consultas que consiste en una instrucción SELECT, son tablas virtuales y están creadas con datos de una o más tablas reales y son presentadas al usuario de una manera real permitiendo ser tratadas igual que una tabla normal.

De igual manera podemos hacer uso de los procedimientos almacenados (**store procedure**), el cual permite reducir el coste de realizar importantes operaciones en la base de datos de una aplicación, un procedimiento almacenado es un programa autocontrolado escrito en lenguaje del DBMS, pueden ser accedidos por cualquier aplicación con el fin de manipular los datos de esta, tiene escritas sus propias sentencias SQL llamadas sentencias Transact SQL, son similares a las funciones o procedimientos en las aplicaciones y son llamados por su nombre, pueden tener parámetros tanto de ingreso de datos, de salida o de ambos.

¿Porque utilizar los procedimientos almacenados?

Existen varios motivos por los cuales utilizar procedimientos almacenados, una de las razones más resaltantes es que resulta más eficiente y más seguro, ya que no se requiere del conocimiento de cómo esta diseñada la base de datos, otros de sus beneficios son:



- La programación modular.
- Distribución.
- Seguridad.
- Ejecución rápida.
- Reducción del tráfico en la red.
- Flexibilidad.

Al igual que desde el ambiente de desarrollo de Visual Studio también podemos crear nuestros reportes desde el ambiente de desarrollo de Crystal Reports y luego ejecutarlos desde nuestro proyecto Visual Basic.Net.

Crystal Reports es el generador de reportes por excelencia de Visual Basic, es un producto creado en su esencia orientados a los usuarios finales, y también posee su propio conjunto de sentencias y funciones para producir información(o sea su propio lenguaje de funciones), por ejemplo permitirle a un ejecutivo crear sus propios informes sin necesidad de asistencia de un desarrollador.



a. Material y Método

Material:

Hardware: tiene las siguientes características.

- Microprocesador Intel Celeron de 2.53 GHz de velocidad
- 40 GB de Disco Duro
- 512 MB de RAM

Software: el software en el que se llevara acabo la aplicación será:

- Microsoft Windows XP Profesional con SP 2 como sistema operativo.
- Microsoft SQL Server 2000 como gestor de bases de Datos.
- Microsoft Visual Basic.Net como lenguaje de programación.
- Crystal Report como generador de informes o reportes.
- Microsoft Office para la elaboración de documentos y diapositivas.



b. Diseño metodológico

Para llevar a cabo la construcción de este sistema utilizamos el método “ciclo de vida clásico” el cual brinda las condiciones adecuadas para el debido almacenamiento y el tratamiento de la información, las etapas a seguir para el desarrollo de este sistema son las siguientes:

Ingeniería y análisis del sistema: En esta etapa se realizaron las debidas entrevistas al cliente del software, en la cual recopilamos la información necesaria para llevar a acabo la construcción del sistema.

Análisis de los requisitos del software: A partir de los datos obtenidos en la primera etapa, logramos comprender el ámbito de dicha información, identificando así las funciones necesarias a utilizar, a si como también definir las interfaces que formaran parte de la estructura de la aplicación. Esta etapa nos proporciona algunas herramientas para llevar a cabo el análisis requerido tales como:

- Diagrama de flujo de datos (DFD)
- Diccionario de datos (DD)
- Especificación de requisitos software (ERS)

Diseño del sistema: En la tercera etapa una vez identificada las funciones y definida la estructura del sistema indicamos como se llevaran a acabo las funciones del sistema y desarrollamos las interfaces que permitirán al usuario comunicarse con el mismo.

Codificación: Para el desarrollo de este sistema hemos hecho uso del entorno de desarrollo de Microsoft Visual Studio.Net (el cual es un entorno de desarrollo integrado que contiene múltiples funcionalidades y todas las herramientas necesarias para



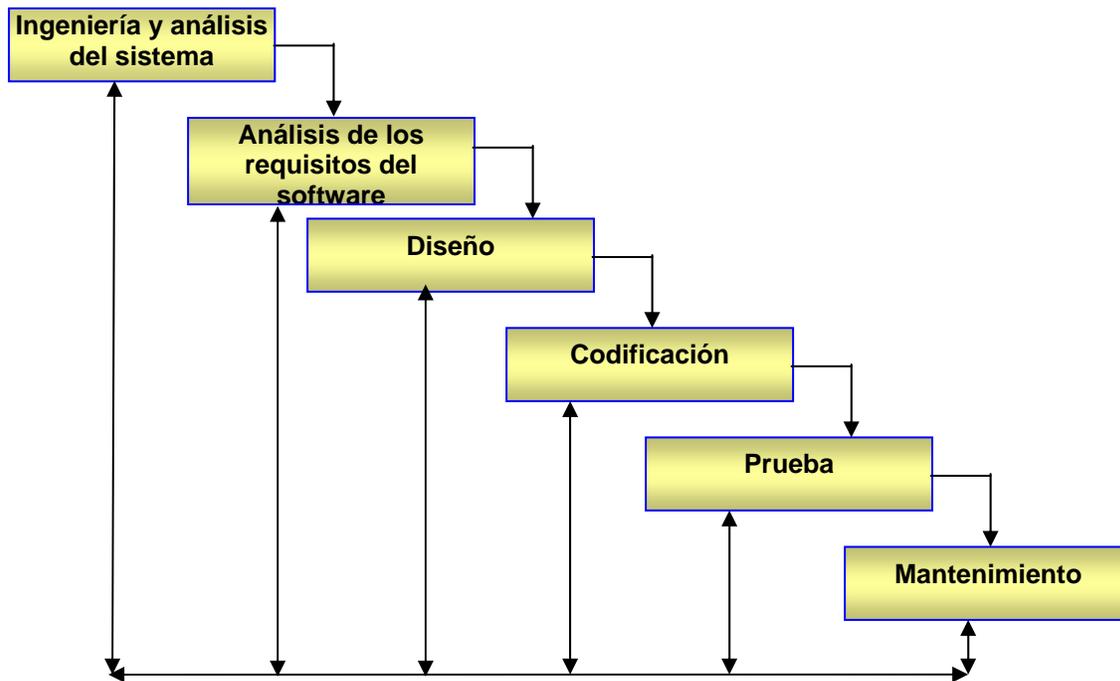
construir programas para Microsoft Windows, es por ello que consideramos que esta plataforma se adapta al desarrollo de nuestra aplicación, también hacemos uso del compilador Microsoft Visual Basic .Net, el cual es un lenguaje orientado a objetos y a eventos y como gestor de base de datos utilizamos Microsoft SQL Server 2000 para la elaboración de las tablas, relaciones, consultas y procedimientos almacenados y Crystal Report para la elaboración de los informes o reportes.

Prueba: Aquí nos centramos en la lógica interna del software, asegurando que todas las sentencias se han probado, y observar que las salidas generadas por el sistema sean las correctas.

Mantenimiento: en esta etapa el administrador del sistema de datos debe dar limpieza periódica de la base de datos, actualización de los mismos logrando de esta forma que los datos con los que trabaja el sistema sean siempre los reales.



c. Estructura del Modelo de Vida Clásico.





Este método también permitirá que el análisis de los requisitos del software se realice mediante las siguientes herramientas, las cuales son de gran utilidad porque permiten una interacción entre el usuario y la aplicación, facilitando el flujo de datos a lo largo del desarrollo del proyecto.

- **Diagrama de Flujo de Datos (DFD)**

Es una técnica gráfica que representa el flujo de información y las transformaciones que se aplican a los datos al moverse desde la entrada hacia la salida.

- **Diccionario de Datos (DD)**

Es un listado organizado donde se detallan todas las definiciones de los elementos del Sistema (flujos de datos, procesos y almacenes de datos).

- **Especificación de Requisitos (ERS)**

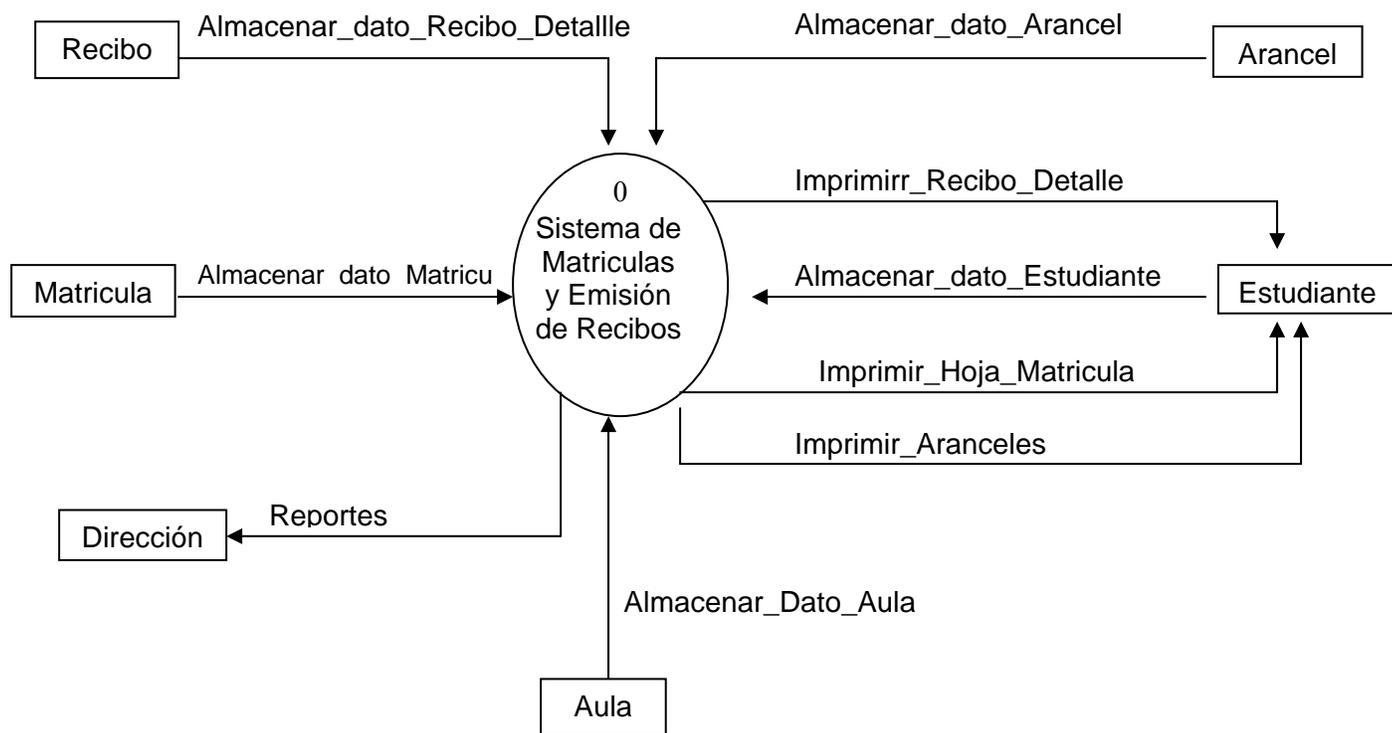
Es el establecimiento conciso de un conjunto de requisitos que deben ser satisfechos por un producto o un proceso, indicando, siempre que sea adecuado, el procedimiento mediante el cual se puede determinar si se han logrado satisfacer los requisitos.



VII. ANALISIS

Descripción del sistema

Proyect Name: Sistema de Matriculas y Emisión de Recibos
Proyect Path: c:\dfd\
Chart File:
Chart Name: Yourdon – Context Diagram
Create On: Dec - 21 - 2006
Create By: Silvia Vanessa Morales
Modified On: Dec - 21 - 2006
Modified By: Silvia Vanessa Morales

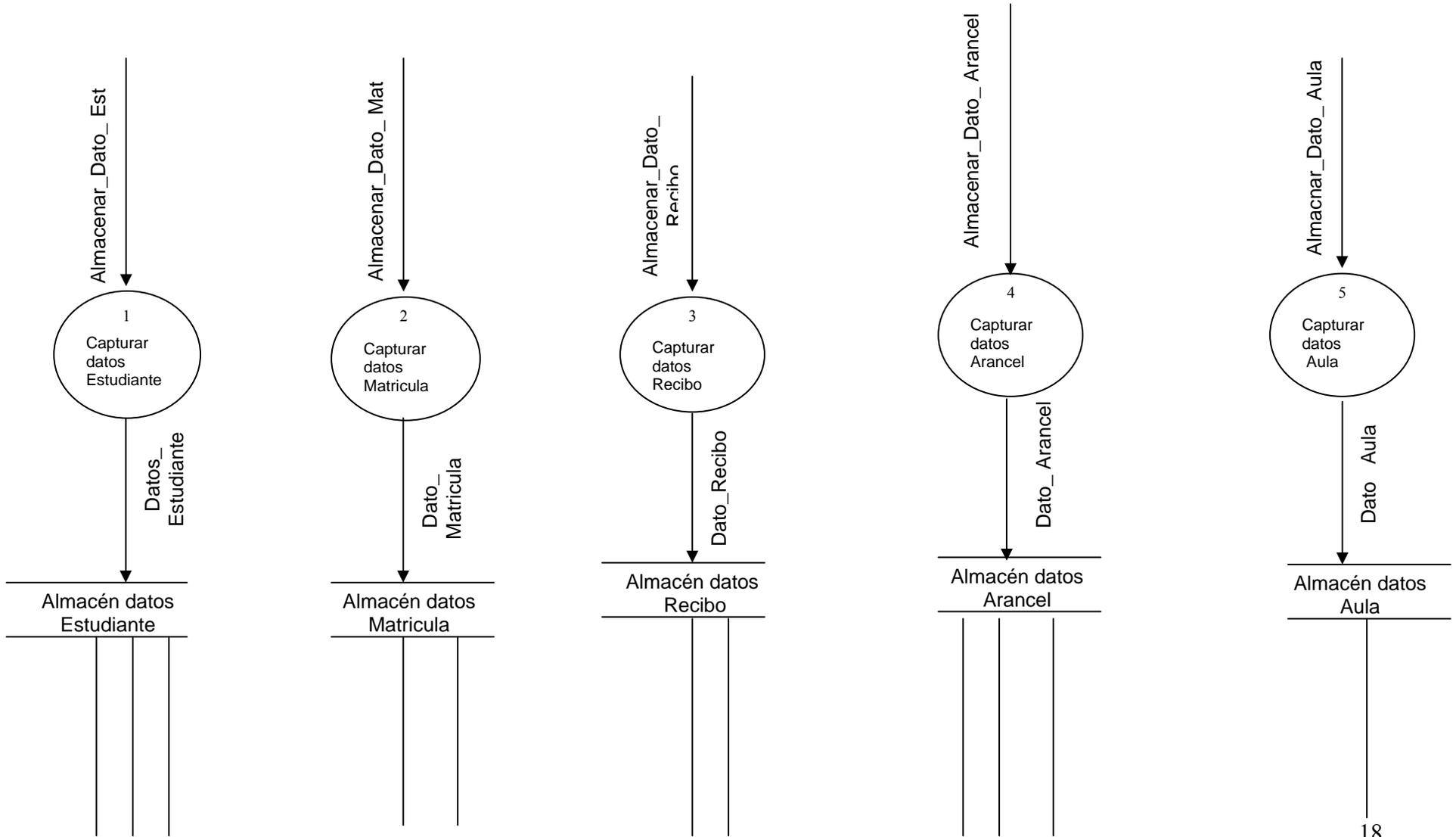


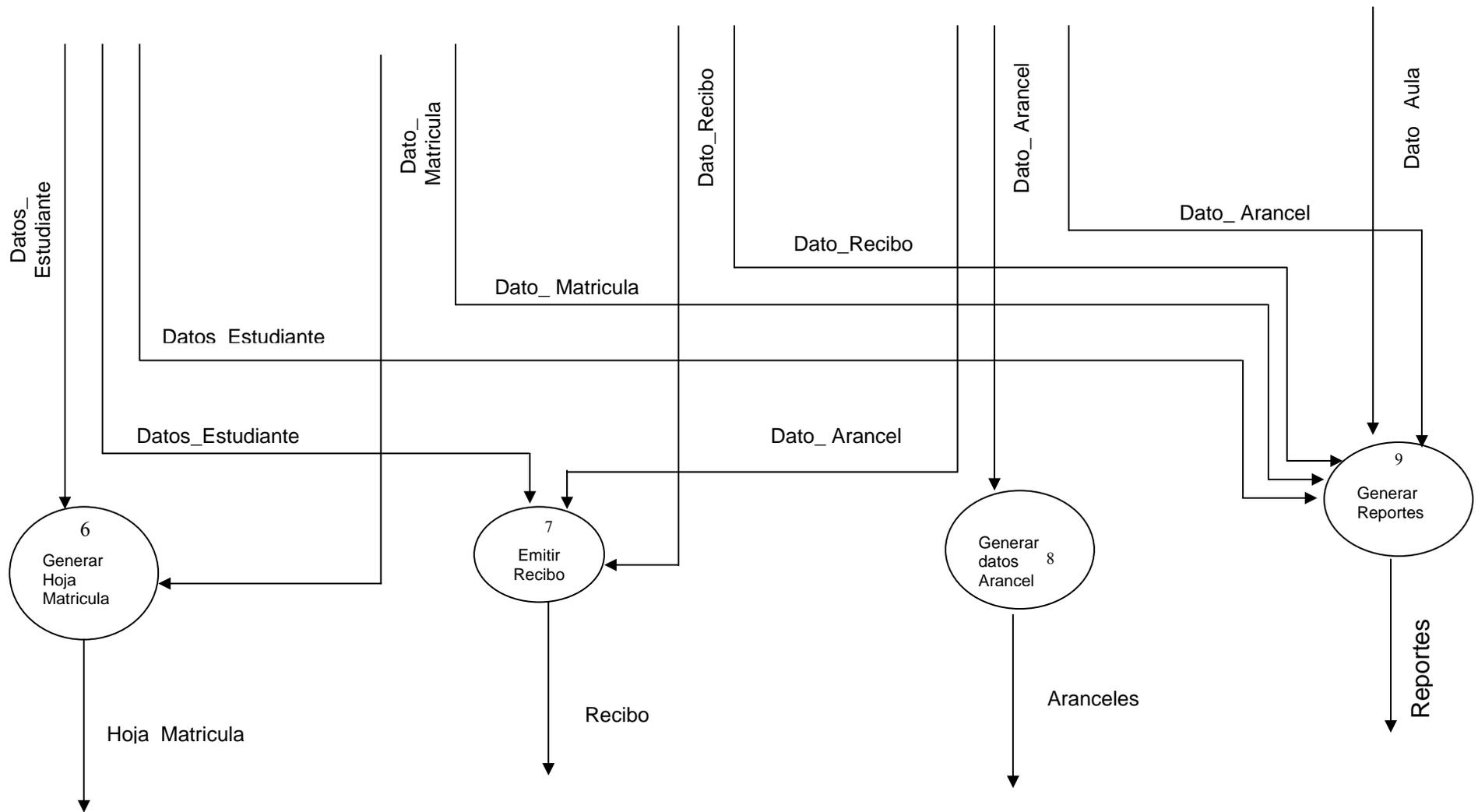
Nivel 0

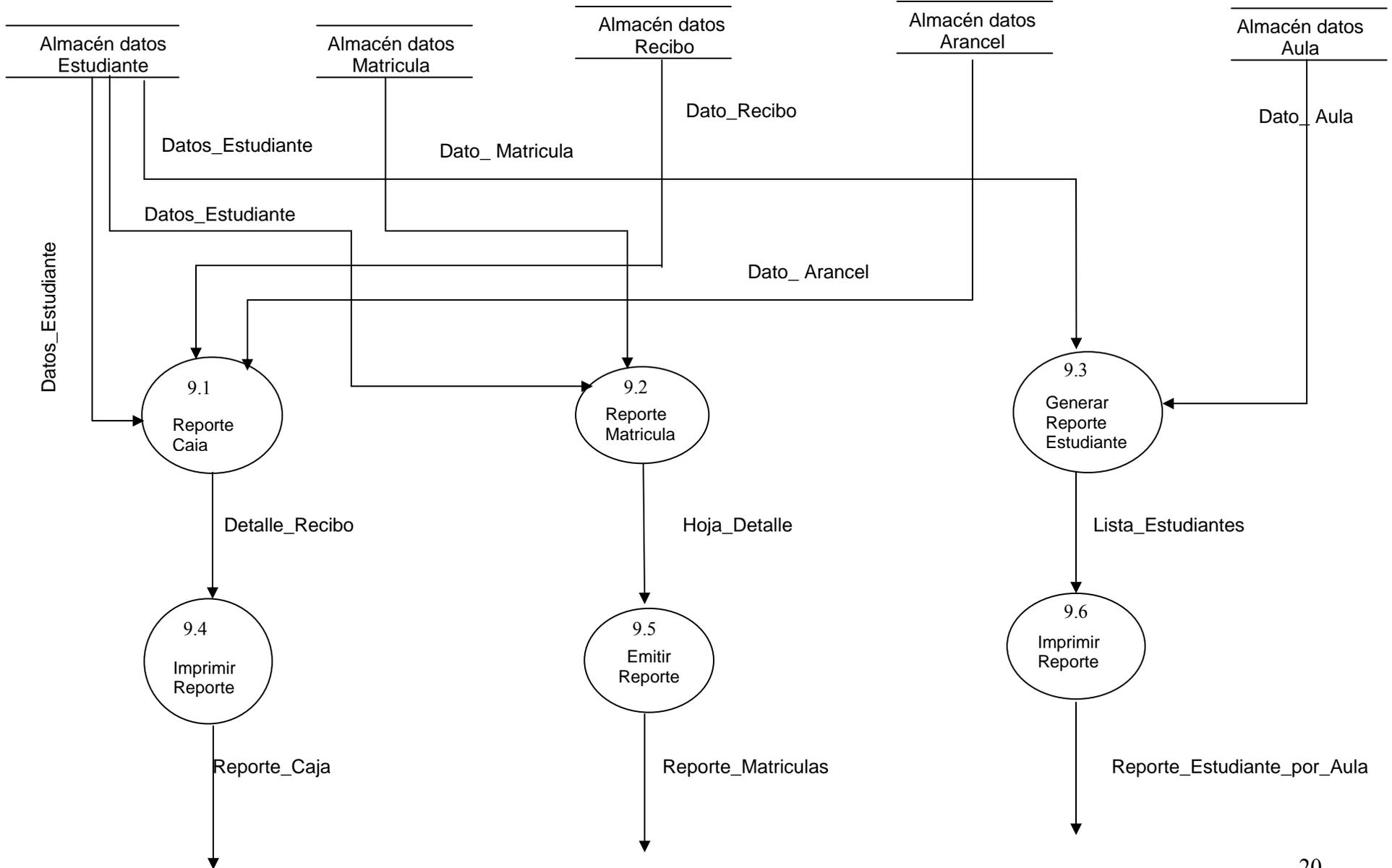
Diagrama de flujo de datos (DFD)



Nivel 1









Diccionario de datos

- **Almacenar los datos de un nuevo ESTUDIANTE**

Nombres + Apellidos + Lugar de nacimiento + Fecha de nacimiento + Edad + Sexo + Tipo de sangre + Dirección + Teléfono

- **Almacenar los datos de una nueva MATRÍCULA**

Código del Estudiante + Código del Aula + Turno + Año lectivo + Grado de la matrícula + Fecha de la matrícula + Partida de nacimiento + Notas del último grado aprobado + Diploma de sexto grado + Otros documentos + Repitente + Observación

- **Almacenar los datos de una nueva AULA**

Nombre + Descripción

- **Almacenar los datos de un nuevo RECIBO Y SU RESPECTIVO DETALLE**

Código del Estudiante + Referencia + Fecha del recibo + Concepto + Total + Valor en letras + Código del Arancel + Descripción + Año pagado + Mes pagado + Sub Total

- **Almacenar los datos de un nuevo Arancel**

Nombre + Costo + Descripción

- **Imprimir Hoja de Matricula**

Código Matricula



- **Imprimir Recibo y su Respectivo Detalle**

Código Recibo

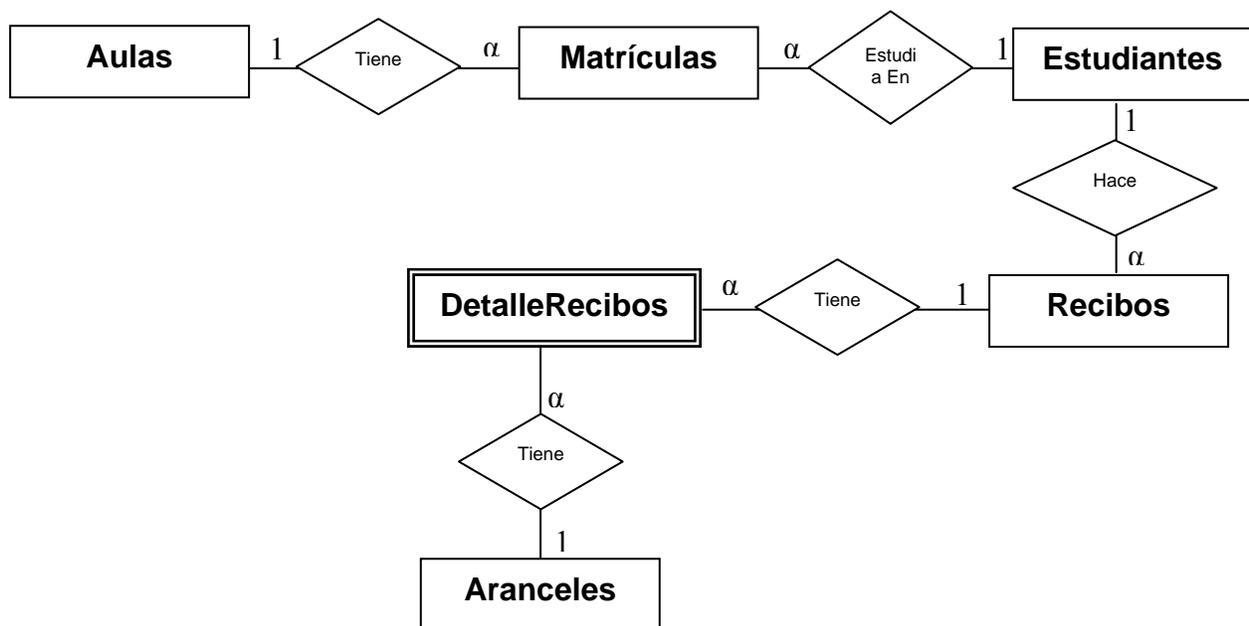
- **Imprimir Listado de Estudiantes por año y por Aula**

Código del Aula

- **Imprimir Estado de Cuenta de un Estudiante**

Año + Identificador del estudiante

Diagrama Entidad – Relación





Especificación de Requisitos del Software

1. Introducción.

1.1. Propósito.

Definir detalladamente los requisitos del software para la creación de **Sistema de Automatización para la Elaboración e Impresión de Recibos y Matrículas del Instituto Socorro Santana Solís (ISSS)**, el cual consiste en la creación de un sistema que permita a los usuarios realizar cualquier procedimiento o consulta de manera fácil, rápida y segura.

1.2. Alcance.

El nombre de la aplicación será: **Sistema de Automatización para la Elaboración e Impresión de Recibos y Matrículas del Instituto Socorro Santana Solís (ISSS)**

Funciones que realizará el software:

1. Almacenar los datos de un nuevo **Estudiante**
2. Almacenar los datos de una nueva **Matrícula**
3. Almacenar los datos de **Aulas**
4. Almacenar los datos de un nuevo **Recibo y su Respectivo Detalle**
5. Almacenar los datos de un nuevo **Arancel**
6. Imprimir **Hoja de Matricula**
7. Imprimir **Recibo y su Respectivo Detalle**
8. Imprimir **Lista de Aranceles**
9. Imprimir **Listado de Estudiantes por Aula**
10. Imprimir **Estado de Cuenta de un Estudiante**



1.3. Definiciones, acrónimos y abreviaturas.

- **Usuario:** Son todos aquellos que podrán hacer uso de la aplicación.
- **Hoja de Matrícula:** documento en el cual aparecerán los datos propios de la matrícula elaborada. En ésta aparecerán datos del estudiante así como la sección y el año que cursa.
- **Recibo:** es un documento que sirve como comprobante de entrada de dinero al centro escolar.
- **Arancel:** es un listado de “cosas” que los estudiantes están obligados a pagar en todo el transcurso del año escolar.
- **Servidor:** Es un ordenador con ciertas características predominantes que mediante un sistema operativo de red, permite administrar y controlar ciertos ordenadores pequeños llamados clientes, con el objetivo de que estos compartan recursos del servidor y puedan al mismo tiempo comunicarse entre si para algunos propósitos.
- **Cliente:** Son ordenadores pequeños que mediante un programa y algunos periféricos son capaces de acceder al servidor para algunos propósitos.
- **Cliente/Servidor:** Permite compartir los recursos de un sistema, distintos de una organización.
- **Red:** Es un conjunto de ordenadores conectados entre si, que permite compartir información y recursos.



1.4. Visión general.

Primeramente se realizará una descripción general del sistema que se desea desarrollar para pasar posteriormente a estudiar cada uno de los requerimientos específicos de forma individual.

2. Descripción general.

2.1. Relaciones del producto

El equipo en el que se desarrollará el producto final es:

- Microprocesador Celeron de 2.53 GHz de velocidad
- 40 GB de Disco Duro
- 512 MB de RAM

El equipo en el que se implementará el producto final es:

1. Máquina servidora de la base de datos:

- Microprocesador Celeron de 2.56 GHz de velocidad
- 80 GB de Disco Duro
- 768 MB de RAM

2. Máquina en la que se instalará el módulo de caja:

- Microprocesador Pentium IV de 1.8 GHz de velocidad
- 40 GB de Disco Duro
- 256 MB de RAM

3. Máquinas en las que se instalará el módulo de matrícula:

- Microprocesador Pentium IV de 1.8 GHz de velocidad
- 40 GB de Disco Duro
- 512 MB de RAM



2.2. Funciones del Sistema.

El sistema realizará las siguientes funciones:

1. Cuando el usuario desee agregar un nuevo **Estudiante**, deberá introducir los datos en el sistema, para que éstos sean almacenados en la base de datos. El código del **Estudiante** se generará automáticamente por el sistema.
2. Cuando el usuario necesite agregar una nueva **Matrícula**, deberá verificar si es de nuevo ingreso o reingreso, si es nuevo ingreso primero deberá agregar los datos del nuevo **Estudiante**, y luego proceder a almacenar los datos propios de la **Matrícula**, de lo contrario el usuario deberá buscar los datos del **Estudiante** en el sistema para hacerle alguna modificación si procede, y posteriormente almacenar los datos de la nueva **Matrícula**. El código de la **Matrícula** será generado automáticamente por el sistema.
3. Cuando el usuario necesite agregar una nueva **Aula**, deberá introducir los datos en el sistema para que éstos sean almacenados en la base de datos. El código de la nueva **Aula** será generado automáticamente por el sistema.
4. Cada vez que el usuario necesite hacer un **Recibo**, primero deberá buscar al estudiante, una vez encontrado deberá comenzar a especificar los datos del recibo y su respectivo detalle, y finalmente almacenar estos datos en la base de datos. El código del **Recibo y su Respectivo Detalle** será generado automáticamente por el sistema.
5. Cuando se vea la necesidad de agregar un nuevo **Arancel**, el usuario deberá introducir los datos en el sistema. El código del **Arancel** será generado automáticamente por el sistema.



6. Cada vez que se matricule a un estudiante en el Instituto, el usuario deberá imprimir una **Hoja de Matrícula** para que ésta sea agregada al expediente del estudiante que el Instituto lleva. También esta **Hoja de Matrícula** se podrá imprimir en cualquier momento siempre y cuando sea a solicitud de la parte interesada (Instituto y/o Estudiante) y tenga justificación para poder hacer esta impresión, el usuario deberá buscar la matrícula que desea imprimir.
7. Cada vez que un estudiante llegue a pagar a caja o bien que el usuario lo estime conveniente, éste podrá **Imprimir el Recibo y su respectivo Detalle** donde se expresa quien pagó, cuanto pagó, y bajo que concepto.
8. Previo a la matrícula o en cualquier momento el usuario puede imprimir una **Lista de Aranceles** para entregársela al estudiante o a su responsable para que de esa manera se sepa cuales son nombres de los **Aranceles** y sus respectivos costos.
9. Cada vez que se necesite, el usuario puede imprimir un listado de **estudiantes por Aula**, de esta manera obtendrán datos personales del estudiantes así como también el año que cursa.
10. Cada vez que se necesite saber sobre los pagos que ha realizado un estudiante en un determinado año el usuario puede perfectamente imprimir un **Estado de Cuenta de** los estudiantes que actualmente se encuentran matriculados en el centro.

2.3. Características del usuario.

Los usuarios finales que van a interactuar con la aplicación serán personas que tienen poco conocimiento informático, motivo por el cual serán capacitados para el



manejo del sistema y también se les proporcionará una ayuda en línea integrada con la aplicación.

2.4. Restricciones generales.

Lenguaje de programación: Microsoft Visual Basic .NET

Gestor de base de datos: Microsoft SQL Server

Generador de reportes: Crystal Report

3. Requisitos específicos.

3.1. Requisitos funcionales.

3.1.1. Almacenar los datos de un nuevo Estudiante

3.1.1.1. Especificación.

3.1.1.1.1. Introducción.

Este proceso realiza la captura de los datos de un nuevo **Estudiante**.

3.1.1.1.2. Entradas.

Por pantalla: el usuario deberá proporcionar los siguientes datos

- Nombres
- Apellidos
- Lugar de nacimiento
- Fecha de nacimiento
- Edad
- Sexo
- Tipo de sangre



- Dirección
- Teléfono

Datos proporcionados por el sistema: referentes al **Estudiante**.

- Código

3.1.1.1.3. Proceso.

Se muestra una pantalla de introducción de datos al usuario. El código de un nuevo **Estudiante** es asignado automáticamente. Posteriormente se actualizará la base de datos con el nuevo registro.

Los datos necesarios y/u obligatorios a introducir serán:

- Nombres
- Apellidos
- Lugar de nacimiento
- Fecha de nacimiento
- Edad
- Sexo
- Tipo de sangre
- Dirección
- Teléfono

A partir de estos datos se registrará un nuevo **Estudiante** y se actualizará la base de datos.



3.1.1.1.4. Salida.

Con todos los datos mencionados anteriormente se almacena en la base de datos el registro del nuevo **Estudiante**.

3.1.2. Almacenar los datos de una nueva Matrícula

3.1.2.1. Especificación.

3.1.2.1.1. Introducción.

Este proceso realiza la captura de los datos de una **Matricula**

3.1.2.1.2. Entradas.

Por pantalla: datos para codificar una nueva **Matricula**

- Código del Estudiante
- Código del Aula
- Turno
- Año lectivo
- Grado de la matrícula
- Fecha de la matrícula
- Partida de nacimiento
- Notas del último grado aprobado
- Diploma de sexto grado
- Otros documentos
- Repitente
- Observación



Datos proporcionados por el sistema: referentes a la **Matrícula**.

- Código

3.1.2.1.3. Proceso.

Se muestra una pantalla de introducción de datos al usuario. El código de una nueva **Matrícula** es asignado automáticamente. Posteriormente se actualizará la base de datos con el nuevo registro.

Los datos necesarios y/u obligatorios a introducir serán:

- Código del Estudiante
- Código del Aula
- Turno
- Año lectivo
- Grado de la matrícula
- Fecha de la matrícula
- Partida de nacimiento
- Notas del último grado aprobado
- Diploma de sexto grado
- Otros documentos
- Repitente
- Observación

A partir de estos datos se registrará una nueva **Matricula** y se actualizará la base de datos.



3.1.2.1.4. Salida.

Con todos los datos mencionados anteriormente se almacena en la base de datos el registro de la nueva **Matricula**.

3.1.3. Almacenar los datos de una nueva Aula

3.1.3.1. Especificación.

3.1.3.1.1. Introducción.

Este proceso realiza la captura de los datos de una **Aula**

3.1.3.1.2. Entradas.

Por pantalla: datos para codificar una nueva **Aula**

- Nombre
- Descripción

Datos proporcionados por el sistema: referentes al **Aula**.

- Código

3.1.3.1.3. Proceso.

Se muestra una pantalla de introducción de datos al usuario. El código de una nueva **Aula** es asignado automáticamente. Posteriormente se actualizará la base de datos con el nuevo registro.



Los datos necesarios y/u obligatorios a introducir serán:

- Nombre
- Descripción

A partir de estos datos se registrará una nueva **Aula** y se actualizará la base de datos.

3.1.3.1.4. Salida.

Con todos los datos mencionados anteriormente se almacena en la base de datos el registro de la nueva **Aula**.

3.1.4. Almacenar los datos de un nuevo Recibo y su Respectivo Detalle

3.1.4.1. Especificación.

3.1.4.1.1. Introducción.

Este proceso realiza la captura de los datos de un **Recibo y su Respectivo Detalle**

3.1.4.1.2. Entradas.

Por pantalla: datos para codificar un nuevo **Recibo**

- Código del Estudiante
- Fecha del recibo
- Concepto
- Código del Arancel



- Descripción
- Año pagado
- Mes pagado

Datos proporcionados por el sistema: referentes al **Recibo** y su **Respectivo Detalle.**

- Código del recibo
- Referencia
- Sub - Total
- Valor total en letras
- Total

3.1.4.1.3. Proceso.

Se muestra una pantalla de introducción de datos al usuario. El código, referencia, el sub total y el valor total en letras del nuevo **Recibo** son proporcionados por el sistema. Posteriormente se actualizará la base de datos con el nuevo registro.

Los datos necesarios y/u obligatorios a introducir serán:

- Código del Estudiante
- Fecha del recibo
- Concepto
- Código del Arancel
- Descripción
- Año pagado



- Mes pagado

A partir de estos datos se registrará un nuevo **Recibo y su respectivo detalle**, y se actualizará la base de datos.

3.1.4.1.4. Salida.

Con todos los datos mencionados anteriormente se almacena en la base de datos el registro del nuevo **Recibo**.

3.1.5. Almacenar los datos de un nuevo Arancel

3.1.5.1. Especificación.

3.1.5.1.1. Introducción.

Este proceso realiza la captura de los datos de un **Arancel**

3.1.5.1.2. Entradas.

Por pantalla: datos para codificar un nuevo **Arancel**

- Nombre
- Costo
- Descripción

Datos proporcionados por el sistema: referentes al **Arancel**.

- Código del Arancel



3.1.5.1.3. Proceso.

Se muestra una pantalla de introducción de datos al usuario. El código de un nuevo **Arancel** es asignado automáticamente. Posteriormente se actualizará la base de datos con el nuevo registro.

Los datos necesarios y/u obligatorios a introducir serán:

- Nombre
- Costo
- Descripción

A partir de estos datos se registrará un nuevo **Arancel** y se actualizará la base de datos.

3.1.5.1.4. Salida.

Con todos los datos mencionados anteriormente se almacena en la base de datos el registro del nuevo **Arancel**.

3.1.6. Imprimir Hoja de Matricula

3.1.6.1. Especificación.

3.1.6.1.1. Introducción.

Este proceso realizará la impresión de los datos de una **Hoja de Matricula**



3.1.6.1.2. Entradas.

Por pantalla: datos para imprimir una **Hoja de Matricula**

- Código de la Matricula

Datos proporcionados por el sistema: referentes a la **Hoja de Matricula.**

- Código del Estudiante
- Nombres del estudiante
- Apellidos del estudiante
- Código del Aula
- Turno
- Año lectivo
- Grado de la matrícula
- Fecha de la matrícula
- Partida de nacimiento
- Notas del último grado aprobado
- Diploma de sexto grado
- Otros documentos
- Repitente
- Observación



3.1.6.1.3. Proceso.

Se muestra una pantalla en la cual el usuario deberá realizar una búsqueda de la matricula a imprimir.

Los datos necesarios y/u obligatorios a introducir serán:

- Código de la Matrícula

A partir de la introducción de este dato se mostrará por pantalla el reporte de la Hoja de Matricula.

3.1.6.1.4. Salida.

- Hoja de Matricula.

3.1.7. Imprimir Recibo y su Respectivo Detalle

3.1.7.1. Especificación.

3.1.7.1.1. Introducción.

Este proceso realizara la impresión de los datos de un **Recibo y su Respectivo Detalle.**

3.1.7.1.2. Entradas.

Por pantalla: datos para imprimir un **Recibo y su Respectivo Detalle.**

- Código del Recibo



Datos proporcionados por el sistema: referentes al **Recibo y su respectivo Detalle**

- Código del recibo
- Código del estudiante
- Nombres del estudiante
- Apellidos del estudiante
- Referencia
- Fecha del recibo
- Concepto
- Código del arancel
- Descripción
- Sub Total
- Año pagado
- Mes pagado
- Total
- Valor total en letras

3.1.7.1.3. Proceso.

Se muestra una pantalla en la cual el usuario deberá realizar una búsqueda del Recibo a Imprimir.

Los datos necesarios y/u obligatorios a introducir serán:

- Código Recibo

A partir de la introducción de este dato se mostrará por pantalla el Recibo y su Respectivo Detalle.



3.1.7.1.4. Salida.

- Hoja de recibo y su respectivo detalle.

3.1.8. Imprimir Lista de Aranceles

3.1.8.1. Especificación.

3.1.8.1.1. Introducción.

Este proceso realizara la impresión de una **Lista de Aranceles**.

3.1.8.1.2. Entradas.

Por pantalla: datos para imprimir una **Lista de Aranceles**.

- Ninguna.

Datos proporcionados por el sistema: referentes a la **Lista de Aranceles**.

- Nombre
- Costo
- Descripción

3.1.8.1.3. Proceso.

El reporte **Lista de Aranceles** es visualizado mediante la ejecución de una consulta.

Los datos necesarios y/u obligatorios a introducir serán:



- Ninguno

3.1.8.1.4. Salida.

- Lista de Aranceles.

3.1.9. Imprimir Listado de Estudiantes por Aula

3.1.9.1. Especificación.

3.1.9.1.1. Introducción.

Este proceso realizará la impresión de un **Listado de Estudiantes por Aula**

3.1.9.1.2. Entradas.

Por pantalla: datos para imprimir un **Listado de Estudiantes por Aula**

- Código del aula

Datos proporcionados por el sistema: referentes al **Listado de Estudiantes por Aula**

- Código del estudiante
- Nombre del estudiante
- Apellidos del estudiante
- Edad



- Sexo
- Teléfono
- Identificador del Aula
- Nombre del Aula
- Grado matrícula

3.1.9.1.3. Proceso.

El usuario realizará una búsqueda por código del aula, posteriormente el reporte **Listado de Estudiantes por Aula** será visualizado por pantalla.

Los datos necesarios y/u obligatorios a introducir serán:

- Código del aula

3.1.9.1.4. Salida.

- Listado de Estudiantes por Aula

Imprimir Estado de Cuenta de un Estudiante

3.1.9.2. Especificación.

3.1.9.2.1. Introducción.

Este proceso realizara la impresión del **Estado de Cuenta de un Estudiante**



3.1.9.2.2. Entradas.

Por pantalla: datos para imprimir el **Estado de Cuenta de un Estudiante**

- Año.
- IdEstudiante

Datos proporcionados por el sistema: referentes al **Estado de Cuenta de un Estudiante**

- Código del estudiante
- Codigo del Recibo
- Referencia
- FechaRecibo
- Codigo del Arancel
- Descripción
- AnyoPagado
- MesPagado
- Nombres del estudiante
- Apellidos del estudiante
- Sub Total
- Total



3.1.9.2.3. Proceso.

El usuario realizará una búsqueda por año y por id del estudiante, posteriormente el reporte **Estado de Cuenta de un Estudiante** será visualizado por pantalla.

Los datos necesarios y/u obligatorios a introducir serán:

- Año.
- IdEstudiante

3.1.9.2.4. Salida.

- Estado de Cuenta de un Estudiante



VIII. DISEÑO

Diseño de Datos

Las siguientes tablas contienen la información de los estudiantes y del procedimiento que este realice, estos serán almacenados cuando el usuario envíe los datos.

Tabla: Estudiantes		
Nombre del campo	Tipo de dato y longitud	Descripción
idEstudiante	Char (8)	Identificador del estudiante
Nombres	Char (250)	Nombres del estudiante
Apellidos	Char (250)	Apellidos del estudiante
LugarNacimiento	Char (250)	El nombre del municipio, departamento o ciudad donde nació el estudiante
FechaNacimiento	Datetime	La fecha en la que el estudiante nació
Edad	Int	Es la edad del estudiante, este dato se calculará a partir de la fecha de nacimiento del estudiante
Sexo	Char (1)	El sexo del estudiante. Solo se almacenará M = Masculino o F = Femenino
TipoSangre	Char (3)	Tipo de sangre del estudiante. Posibles valores: O+, O-, A+, A-, B+, B-, AB+ o AB-
Dirección	Char (500)	La dirección actual de donde vive el estudiante
Teléfono	Char (8)	Teléfono actual del estudiante ya sea convencional o celular



Tabla: Matriculas		
Nombre del campo	Tipo de dato y longitud	Descripción
idMatricula	Char (8)	El identificador de la matrícula
idEstudiante	Char (8)	Identificador del estudiante
idAula	Char (4)	Identificador del aula donde estudiará el estudiante
Turno	Char(10)	Turnos disponibles en el centro (Matutino y Vespertino)
AnyoLectivo	Int	Año lectivo de la matrícula, ejemplo: 2006, 2007, ..., 200n
GradoMatricula	Char (5)	Es el grado en el que se matriculó el estudiante, ejemplo: I, II, III, IV y V año
FechaMatricula	Datetime	Es la fecha en la que se realizó la matrícula
PartidaNacimiento	Char (2)	Si trajo o no partida de nacimiento
NotasUltimoGrado	Char (2)	Si trajo o no las notas del último grado aprobado
DiplomaSextoGrado	Char (2)	Si trajo o no el diploma de sexto grado
OtrosDocumentos	Char (500)	Nombre de algún otro documento que el estudiante o su responsable haya traído
Repitente	Char (2)	Si el estudiante es no repitente
Observación	Char (1000)	Si hay o no alguna observación que hacerle a la matrícula

Tabla: Aulas		
Nombre del campo	Tipo de dato y longitud	Descripción
idAula	Char (4)	El identificador del aula
Nombre	Char (250)	Nombre del aula (I Año A, I Año B, etc)
Descripción	Char (250)	Breve descripción del aula, por ejemplo: color, dimensión, ubicación, estado, etc



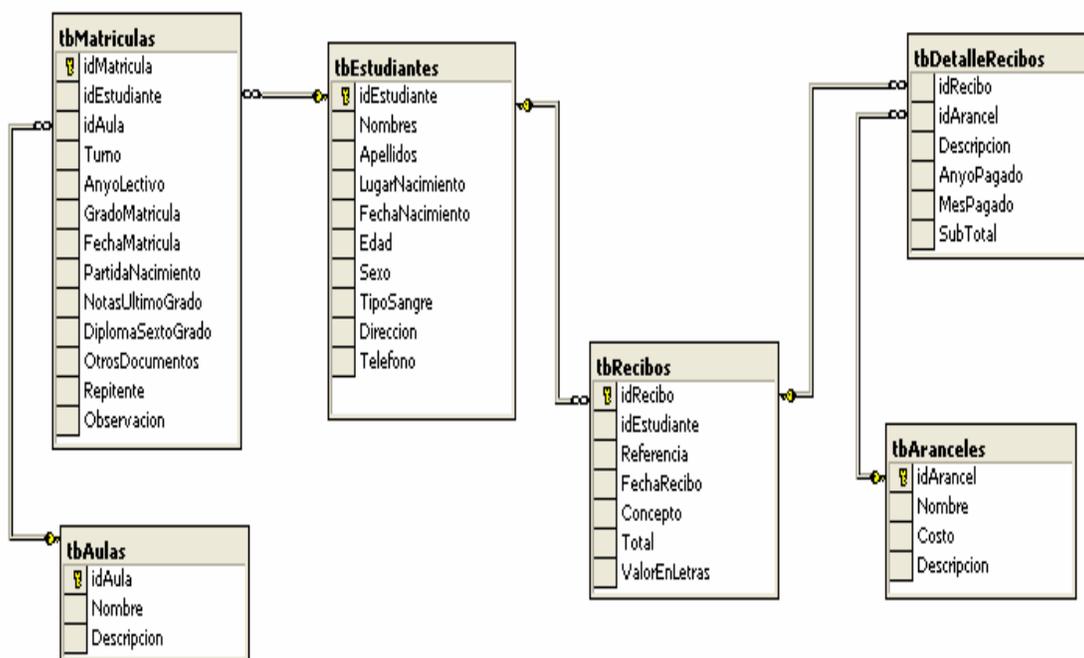
Tabla: Recibos		
Nombre del campo	Tipo de dato y longitud	Descripción
idRecibo	Char (8)	El identificador del recibo
idEstudiante	Char (8)	El identificador del estudiante
Referencia	Char (8)	Cadena de número para especificar una referencia al recibo
FechaRecibo	Datetime	Es la fecha en la que se realizó el recibo
Concepto	Char (1000)	Es el concepto bajo el cual se realizó el recibo
Total	Money	Es el total a pagar por un determinado concepto.
ValorEnLetras	Char (1000)	Es el valor total del recibo pero en letras

Tabla: DetalleRecibos		
Nombre del campo	Tipo de dato y longitud	Descripción
idRecibo	Char (8)	El identificador del detalle del recibo
IdArancel	Char(4)	El identificador de un Arancel
Descripción	Char (500)	Breve descripción de un detalle del recibo.
AnyoPagado	Int	Es el año en el que está pagando, ejemplo: 2006, 2007, 2008,..., 200n
MesPagado	Int	Es el mes en que está pagando o bien el mes de colegiatura a pagar.
SubTotal	Money	Es el valor monetario que tiene asociado cada uno de los aranceles

Tabla: Aranceles		
Nombre del campo	Tipo de dato y longitud	Descripción
idArancel	Char (4)	El identificador del arancel
Nombre	Char (500)	Nombre del arancel
Costo	Money	Es el costo que deben de pagar por este arancel
Descripción	Char (500)	Breve descripción del arancel

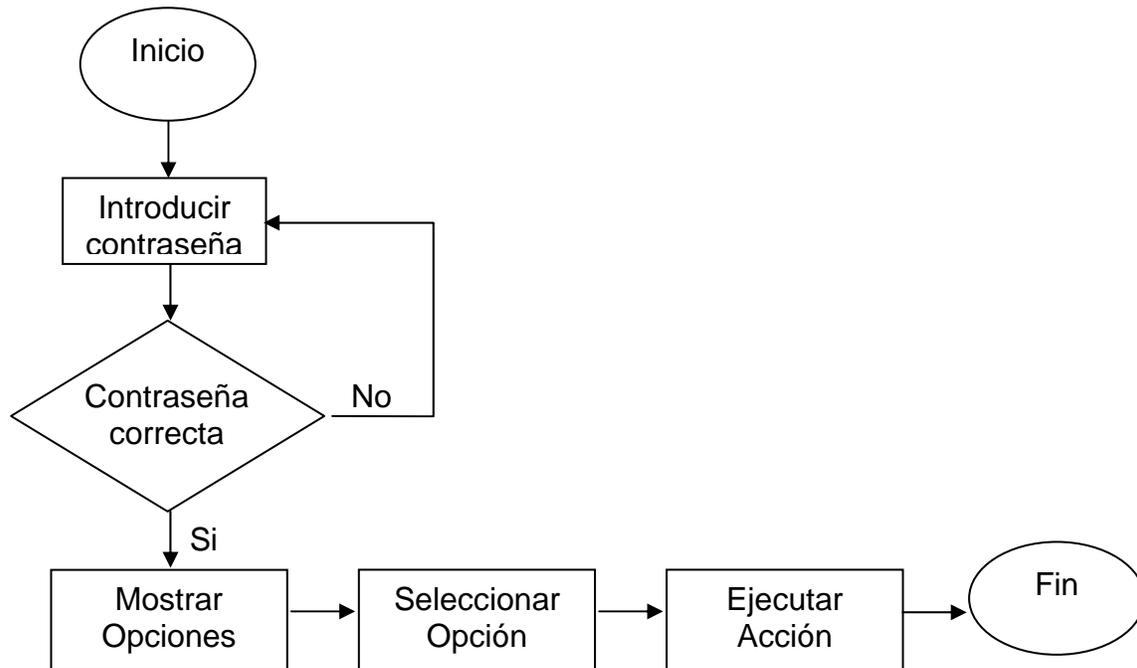


Diseño Relacional



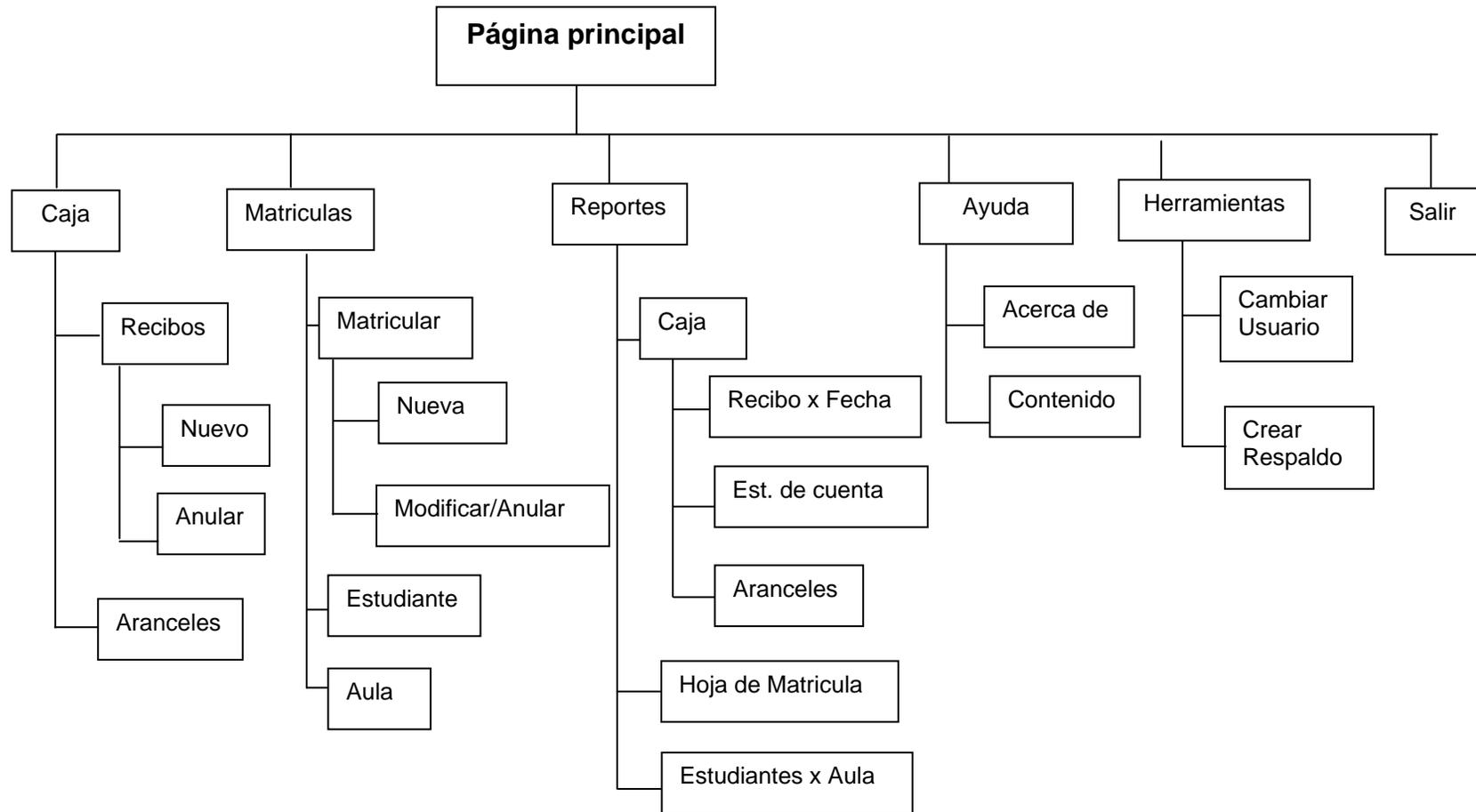


Diseño Procedimental





Diseño Arquitectónico





X. CONCLUSION

La Elaboración e Impresión de Recibos y Matriculas, es sin duda, una actividad que nos puede hacer, mucho más fácil y rápido cualquier gestión que el Estudiante desee realizar, es por ello, que consideramos que es sumamente importante poner atención en el mismo si se desea crear procedimientos que lleven a mejorar la calidad del servicio que ofrecen a sus usuarios.

Luego de haber tomado en cuenta todos los detalles que el cliente nos proporcionó, constatamos que este sistema satisface todas las exigencias tanto en la Elaboración como en la Impresión de Recibos y Matriculas.

En resumen, podemos concluir que el “**Sistema de Automatización para la Elaboración e Impresión de Recibos y Matrículas del Instituto Socorro Santana Solís (ISSS)**”, se ha realizado tomando en cuenta las fases de análisis y diseño con el objetivo de proporcionarles a los usuarios una manera sencilla y fácil de realizar cualquier gestión que se les presente.



RECOMENDACIONES

Una vez que el sistema esté en funcionamiento es recomendable crear copias de respaldos de la información para evitar pérdidas de datos, realizando una revisión periódica para que este pueda ejecutar las operaciones con una mayor rapidez y eficiencia proporcionando así una mayor seguridad a los usuarios (esta copia de respaldo la hará el administrador de la base de datos el cual nos solicitó no agregar esta opción en el sistema ya que él lo hará desde el Sistema Gestor de Base de Datos Ms. SQL Server).

También es recomendable mantener actualizada la fecha del sistema puesto que es utilizada por el sistema en muchos procesos como por ejemplo: en la elaboración de un nuevo recibo y en la generación de algunos reportes.

- Mantener en completo sigilo las cuentas de usuarios que pueden utilizar el sistema.
- Tener mucho cuidado al momento de anular algún recibo o de retirar alguna matrícula.



XI. BIBLIOGRAFÍA

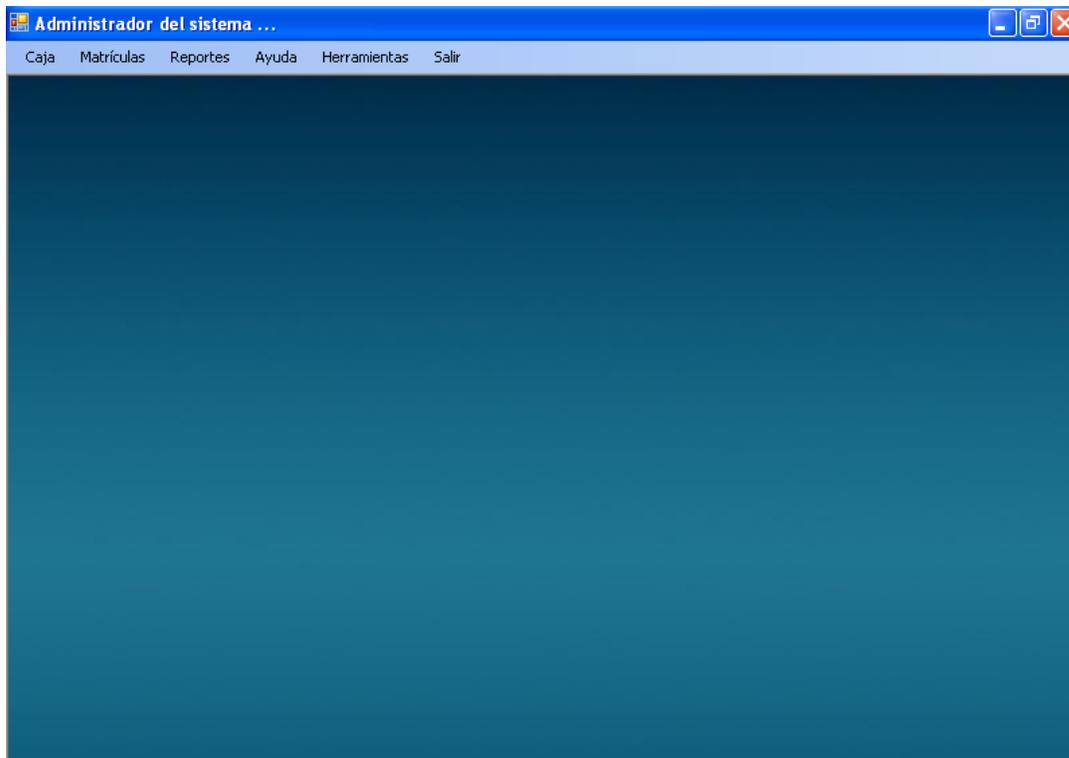
- **Control de compra y venta de productos farmacéuticos de la farmacia ICRIS**
Lic. Carlos Enrique Gómez Mayorga
Lic. Ana Maribel Palma Romero
Lic. Lavinia Alejandra Quiroz Roque
Biblioteca Central.
- **Recursos en Internet**
 - ✚ <http://www.elguille.info>
 - ✚ <http://es.gotdotnet.com/quickstart/aspplus/>
 - ✚ <http://www.support.microsoft.com>



XII. ANEXOS

a. Diseño de Interfaz

Formulario Principal



Formulario Clave

Autenticación de usuarios

Usuario

Clave



Formulario Presentación



Formulario Nuevo Recibo

Nuevo Recibo ...

Id. Recibo: 00000014 Concepto: pago de mensualidades

Referencia: 00000014

Id. Estudiante: 00000003 Buscar

Fecha: 22/02/2007

Id. Arancel: 0007 Buscar Año pagado: 2007 Agregar

Descripción: Pago de colegiatura Mes pagado: 02 Quitar

Sub Total: 10.0000

Id. Recibo	Id. Arancel	Descripción	Año pagado	Mes pagado	Sub Total	
00000014	0007	Pago de colegiatura	...	2007	02	10.0000

Valor en letras: DIEZ CON 0/100

Total C\$: 10

Guardar Cancelar



Formulario Anular Recibo

Anular Recibo ...

Especifique la fecha: 08/02/2007

Recibos encontrados

Recibo	Estudiante	Referencia	Fecha	Concepto	Total	Valor e
00000001	00000001	00000001	08/02/2007	Pago del mes de enero ...	10.0000	Cero
00000002	00000002	00000002	08/02/2007	Pago de los meses de enero y febrer...	20.0000	Cero

Detalle de recibos encontrados

Id. Recibo	Id. Arancel	Descripción	Año pagado	Mes pagado	Sub Total
00000002	0007	Pago de colegiatura	2007	1	10.0000
00000002	0007	Pago de colegiatura	2007	2	10.0000

Formulario Catalogo de Aranceles

Catálogo de Aranceles...

Especifique el nombre del arancel

Id	Nombre	Costo	Descripción
0001	Colegiatura	50.0000	Este es el pago de las mensualidad
0002	Matricula	100.0000	Es el pago de la matricula
0003	Insignea de tela	10.0000	Son las insignias bordadas
0004	Camisetas	25.0000	Es el pago de camisetas



Formulario Nuevo Arancel

Nuevo Arancel...

Identificador

Nombre

Costo

Descripción

Aceptar Cancelar

Formulario Modificar Aranceles

Modificar Aranceles ...

Identificador

Nombre

Costo

Descripción

Aceptar Cancelar



Formulario Nueva Matricula

Nueva Matrícula ...

Id. Matrícula	<input type="text" value="00000011"/>	Otros doc.	<input type="text" value="Ninguno"/>
Id. Estudiante	<input type="text" value="00000001"/> <input type="button" value="Buscar"/>		
Id. Aula	<input type="text" value="0005"/> <input type="button" value="Buscar"/>		
Turno	<input type="text" value="Matutino"/>		
Año lectivo	<input type="text" value="2007"/>	Observación	<input type="text" value="Ninguna"/>
Año	<input type="text" value="I"/>		
Fecha matrícula	<input type="text" value="13/02/2007"/>		
Repitente	<input type="text" value="Si"/>	<input type="button" value="Aceptar"/>	<input type="button" value="Cancelar"/>
Partida Nac.	<input type="text" value="Si"/>		
Notas último grado	<input type="text" value="Si"/>		
Dip. 6to. grado	<input type="text" value="Si"/>		

Formulario Matriculas X Fecha

Matriculas X Fecha ...

Especifique los parámetros

Fecha Inicial: Fecha Final:

Seleccione la acción

Modificar matrícula

Anular/Retirar matrícula

Id. Matrícula	Id. Estudiante	Id. Aula	Turno	Año Lec...	Grado Matrícula	Fecha Matrícula
00000001	00000001	0001	Vespertino	2007	I	27/02/2007
00000002	00000002	0001	Matutino	2007	I	27/02/2007
00000003	00000003	0001	Matutino	2007	I	27/02/2007
00000004	00000004	0001	Matutino	2007	I	27/02/2007
00000005	00000005	0002	Matutino	2007	I	27/02/2007
00000006	00000006	0002	Matutino	2007	I	27/02/2007
00000007	00000007	0002	Matutino	2007	I	27/02/2007
00000008	00000008	0003	Matutino	2007	I	27/02/2007
00000009	00000009	0003	Matutino	2007	I	27/02/2007
00000010	00000010	0004	Matutino	2007	I	27/02/2007
00000011	00000011	0004	Matutino	2007	I	27/02/2007
00000012	00000012	0005	Matutino	2007	I	27/02/2007
00000013	00000013	0005	Matutino	2007	I	27/02/2007
00000014	00000014	0005	Matutino	2007	I	27/02/2007
00000015	00000015	0006	Matutino	2007	I	27/02/2007



Formulario Modificar Matriculas

Modificar Matriculas ...

Id. Matrícula: Otros doc.:

Id. Estudiante:

Id. Aula:

Turno:

Año lectivo:

Año:

Fecha matrícula:

Observación:

Repitente:

Partida Nac.:

Notas último grado:

Dip. Bto. grado:

Formulario Catalogo de Estudiantes

Catálogo de Estudiantes ...

Especifique los apellidos del estudiante:

Id. Est.	Nombres	Apellidos	Lugar Nac.	Fecha Nac.	Edad	Sexo
00000001	Kemberling Yulieth	Aguirre Santamaria	León	27/04/1990	17	M
00000002	Eduardo Ezequiel	Centeno Belly	León	27/02/1990	17	M
00000003	Erick Mauricio	Diaz Hernandez	Corinto	27/02/1985	27	M
00000004	Katherine Nahima	Donaire Vallecillo	El Viejo	27/02/1985	27	F
00000005	Roberto Carlo	Fernandez Montoya	Estelí	27/02/1980	27	M
00000006	Jainer Antonio	Fonseca Parajon	Managua	27/02/1990	17	M
00000007	Karen Samanta	Hernandez Pereira	Granada	27/02/1997	10	F
00000008	Fania Leon	Meneses Peraza	Chinandega	27/02/1997	10	F
00000009	Marvin Antonio	Meza Hernandez	León	27/02/1994	11	M
00000010	Edwin Jose	Navas Alvarado	Telica	27/02/1994	11	M
00000011	Karen Magdalena	Rodriguez Martinez	Quezalguaque	27/02/1993	12	F
00000012	Yeriel Francis	Acosta Diaz	Telica	27/02/1992	13	F
00000013	Jorge Rene	Arauz Espinoza	Telica	27/02/1992	15	M
00000014	Aryana de los Angeles	Garay Martinez	León	28/02/1992	15	F
00000015	Gustavo Ariel	Guerreo Granados	León	28/02/1993	14	M
00000016	Jusiel Regian	Lanzas Hernandez	León	26/03/1991	16	M
00000017	Flor de Maria	Ordoñez Renteria	León	12/03/1991	16	F
00000018	Delvis Antonio	Gavarrete Fajardo	Telica	01/03/1991	16	M
00000019	Ingrid Maria	Davila Rocha	León	27/06/2001	16	F
00000020	Augusta Cesar	Anduray	León	27/06/1990	17	M
00000021	Carlos	Galeano	León	15/06/1990	17	M



Formulario Nuevo Estudiante

Nuevo Estudiante ...

Identificador	00000013	Sexo	M
Nombres		T. Sangre	A+
Apellidos		Dirección	
Lugar de nac	Managua	Teléfono	
Fecha de nac	13/02/2007		
Edad			

Aceptar Cancelar

Formulario Modificar Estudiantes

Modificar Estudiantes ...

Identificador	00000011	Sexo	F
Nombres	Ingrid María	T. Sangre	A-
Apellidos	Flores Delgado	Dirección	Igl. Sn Juan 1 al norte 1/2 c arriba
Lugar de nac	León	Teléfono	-
Fecha de nac	Martes, 05 de Junio		
Edad	22		

Aceptar Cancelar



Formulario Catalogo de Aulas

Catálogo de Aulas...

Especifique el nombre del aula

Id	Nombre	Descripción
0001	I Año A	Primer año
0002	I Año B	Primer año
0003	II Año A	Segundo año
0004	III Año A	Tercer año
0005	III Año B	Tercer año
0006	IV Año A	Cuarto año
0007	IV Año B	Cuarto año
0008	V Año A	Quinto año
0009	V Año B	Quinto año
0010	V Año C	Quinto año
0011	VI	sexto año

Formulario de Aulas

Aulas...

Identificador

Nombre

Descripción



Formulario Modificar Aulas

Modificar Aulas ...

Identificador: 0008

Nombre: V Año A

Descripción: Quinto año

Aceptar Cancelar

Formulario Recibos X Fecha

Recibos X Fecha ...

Especifique la fecha: 08/02/2007

Recibos encontrados

Recibo	Estudiante	Referencia	Fecha	Concepto	Total	Valor e
00000001	00000001	00000001	08/02/2007	Pago del mes de enero ...	10.0000	Cero
00000002	00000002	00000002	08/02/2007	Pago de los meses de enero y febrer...	20.0000	Cero

Detalle de recibos encontrados

Id. Recibo	Id. Arancel	Descripción	Año pagado	Mes pagado	Sub Total
00000002	0007	Pago de colegiatura	2007	1	10.0000
00000002	0007	Pago de colegiatura	2007	2	10.0000

Imprimir Cerrar



Formulario Mostrar Recibo

Mostrar Recibo ...

95% 1 of 1

Preview

Comprobante de Ingreso a Caja

Ho. Recibo: 00000002
Referencia: 00000002

Fecha: Jueves, Febrero 8, 2007

Recibi de: 00000002 Morales Maradiaga
María del Carmen

La cantidad de: *Cero*
En concepto de: Pago de los meses de enero y febrero

idArancel	Arancel	Año	Mes	SubTotal
0007	Pago de colegiatura	2,007	1	C\$ 10.00
0007	Pago de colegiatura	2,007	2	C\$ 10.00
Total del recibo:				C\$ 20.00

Maria del Carmen Morales M.
Cajera

Formulario Estado de Cuenta

Estado de cuenta ...

Especifique el Año

Id. Estudiantes



Formulario Reporte de Estado de Cuenta

Estado de cuenta por estudiante, emitido el: 13/02/2007

idEstudiante: 00000002
Nombres: María del Carmen
Apellidos: Morales Maradiaga

<u>Id. Recibo</u>	<u>Referencia</u>	<u>Fecha</u>	<u>Id. Arancel</u>	<u>Descripción</u>	<u>Año</u>	<u>Mes</u>	<u>Sub Total</u>
00000002	00000002	08/02/2007	0007	Pago de colegiatura	2,007	Enero	C\$ 10.00
00000002	00000002	08/02/2007	0007	Pago de colegiatura	2,007	Febrero	C\$ 10.00
Total:							C\$ 20.00

Formulario CambiarUsuario

Cambio de usuario ...

Usuario

Clave



d. Codificación:

frmAutenticacion:

```
Public Class frmAutenticacion
```

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

```
End
```

```
End Sub
```

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

```
Call modUtilidades.DesactivarBarraDeMenu()
```

```
End Sub
```

```
Sub ValidarUsuario()
```

```
Dim Conexion As String = "Data Source=PERSONAL;initial catalog=bdINSS;Trusted_Connection=False;packet size=4096;uid=" + Trim(Me.txtUsuario.Text) + ";" + "pwd=" + Trim(Me.txtClave.Text) + ";"
```

```
Try
```

```
If Len(Me.txtUsuario.Text) = 0 Or Len(Me.txtClave.Text) = 0 Then  
    MsgBox("Debe especificar un usuario y una clave", MsgBoxStyle.Critical, "ISSS")
```

```
Exit Sub
```

```
Elseif Me.txtUsuario.Text = "sa" Then
```

```
    MsgBox("Acceso denegado para este usuario", MsgBoxStyle.Critical, "ISSS")
```

```
Exit Sub
```

```
Else
```

```
Dim cn As New Data.SqlClient.SqlConnection(Conexion)  
cn.Open()
```

```
If cn.State = 1 Then
```

```
    MsgBox("Conexion establecida ...")
```

```
    modUtilidades.usrAux = Me.txtUsuario.Text.ToLower()
```

```
    modUtilidades.claveAux = Me.txtClave.Text.ToLower()
```

```
    Me.Hide()
```



```
Me.Close()  
frmPresentacion.Show()  
cn.Close()  
cn = Nothing  
End If  
End If
```

```
Catch ex As Exception  
If Err.Number = 5 Then  
    MsgBox("Usuario o Clave incorrecto", MsgBoxStyle.Critical, "ISSS")  
Else  
    MsgBox("Error ocurrido: " + Err.Description)  
End If
```

```
End Try
```

```
End Sub
```

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

```
    Call Me.ValidarUsuario()
```

```
End Sub
```

```
End Class
```

frmPresentacion

```
Public Class frmPresentacion
```

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

```
        Me.Hide()  
        Me.Close()
```

```
        modUtilidades.DesactivarActivarMenu()
```

```
End Sub
```

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



```
Call modUtilidades.DesactivarBarraDeMenu()
```

```
End Sub
```

```
End Class
```

Codigo fuente del formulario frmNuevoRecibo:

```
Public Class frmNuevoRecibo
```

```
' Esta var la vamos a utilizar para controlar  
' cuantas filas llevamos agregadas al  
' detalle de los recibos
```

```
Public Fila As Integer = 0
```

```
Sub UltimoCodigoRecibo_Referencia()
```

```
Dim CadenaConexion As String
```

```
CadenaConexion = modUtilidades.Conexion
```

```
Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
```

```
Dim vUltimoCodigo As New Data.SqlClient.SqlCommand("proc_GenerarIdRecibo",
```

```
cn)
```

```
vUltimoCodigo.CommandType = CommandType.StoredProcedure
```

```
Dim drUltimoCodigo As Data.SqlClient.SqlDataReader
```

```
cn.Open()
```

```
drUltimoCodigo = vUltimoCodigo.ExecuteReader
```

```
If drUltimoCodigo.Read Then
```

```
Me.txtIdRecibo.Text = Format(drUltimoCodigo("MaxCodigo"), "00000000")
```

```
Me.txtReferencia.Text = Format(drUltimoCodigo("MaxCodigo"), "00000000")
```

```
Else
```

```
Me.txtIdRecibo.Text = Format("1", "00000000")
```

```
Me.txtReferencia.Text = Format("1", "00000000")
```

```
End If
```

```
'Cerrando objetos
```

```
drUltimoCodigo.Close()
```

```
cn.Close()
```

```
End Sub
```

```
Sub CrearListViewDetalleR()
```

```
'Formatear el control ListView
```

```
With Me.lvDetalleRecibo
```



```
.Clear()  
.LabelEdit = False  
.AllowColumnReorder = True  
.FullRowSelect = True  
.Sorting = SortOrder.Ascending  
.View = View.Details  
.GridLines = True  
.Columns.Add("Id. Recibo", 75, HorizontalAlignment.Center)  
.Columns.Add("Id. Arancel", 75, HorizontalAlignment.Center)  
.Columns.Add("Descripción", 230, HorizontalAlignment.Center)  
.Columns.Add("Año pagado", 75, HorizontalAlignment.Center)  
.Columns.Add("Mes pagado", 75, HorizontalAlignment.Center)  
.Columns.Add("Sub Total", 80, HorizontalAlignment.Center)
```

End With

End Sub

Sub CrearListViewEstudiante()

'Formatear el control ListView

With Me.lvEstudiantes

```
.Clear()
```

```
.LabelEdit = False
```

```
.AllowColumnReorder = True
```

```
.FullRowSelect = True
```

```
.Sorting = SortOrder.Ascending
```

```
.View = View.Details
```

```
.GridLines = True
```

```
.Columns.Add("Id. Est.", 70, HorizontalAlignment.Center)
```

```
.Columns.Add("Nombres", 150, HorizontalAlignment.Center)
```

```
.Columns.Add("Apellidos", 150, HorizontalAlignment.Center)
```

End With

End Sub

Sub CargarEstudiantes()

```
Dim CadenaConexion As String
```

```
CadenaConexion = modUtilidades.Conexion
```

```
Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
```

```
Dim vListarCargos As New Data.SqlClient.SqlCommand("proc_ListarEstudiantes",
```

cn)

```
vListarCargos.CommandType = CommandType.StoredProcedure
```

```
Dim drListarCargos As Data.SqlClient.SqlDataReader
```



```
cn.Open()  
drListarCargos = vListarCargos.ExecuteReader
```

```
'Llenar el ListView
```

```
Dim vpos As Integer
```

```
Me.lvEstudiantes.Items.Clear()
```

```
While drListarCargos.Read
```

```
With Me.lvEstudiantes
```

```
.Items.Add(drListarCargos("idEstudiante")) ' Identificador
```

```
.Items(vpos).SubItems.Add(drListarCargos("Nombres")) ' Nombres
```

```
.Items(vpos).SubItems.Add(drListarCargos("Apellidos")) ' Apellidos
```

```
' Llenar el combo con los nombres de las aulas
```

```
Me.cboApellidosEstudiantes.Items.Add(Trim(drListarCargos("Apellidos"))) '
```

```
Apellidos
```

```
vpos += 1
```

```
End With
```

```
End While
```

```
If Me.cboApellidosEstudiantes.Items.Count = 0 Then
```

```
Me.cmdAceptarIdEstudiante.Enabled = False
```

```
Me.cboApellidosEstudiantes.Enabled = False
```

```
Me.cmdCancelarIdEstudiante.Focus()
```

```
Else
```

```
Me.cmdAceptarIdEstudiante.Enabled = True
```

```
Me.cmdCancelarIdEstudiante.Enabled = True
```

```
Me.cboApellidosEstudiantes.Enabled = True
```

```
Me.cboApellidosEstudiantes.SelectedIndex = 0
```

```
Me.cmdAceptarIdEstudiante.Focus()
```

```
End If
```

```
'Cerrando objetos
```

```
drListarCargos.Close()
```

```
cn.Close()
```

```
End Sub
```

```
Sub EstudiantesXApellidos()
```

```
Dim CadenaConexion As String
```

```
Dim f As Integer
```

```
Try
```

```
CadenaConexion = modUtilidades.Conexion
```

```
Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
```



```
Dim ocmd As SqlConnection.SqlCommand = New
SqlConnection.SqlCommand("proc_EstudiantesXApellidos", cn)
Dim oParametros As New SqlConnection.SqlParameter()
'Apellidos del estudiante a buscar
If Me.cboApellidosEstudiantes.Text = "" Then
    Call Me.CargarEstudiantes()
Else
    oParametros = ocmd.Parameters.Add("@apellidos", SqlDbType.Char, 250)

    oParametros.Value = Me.cboApellidosEstudiantes.Text
    ocmd.CommandType = CommandType.StoredProcedure
    cn.Open()
    Dim oda As New Data.SqlClient.SqlDataAdapter(ocmd)

    Dim odt As New DataTable()
    oda.Fill(odt)
    ' Creamos un objeto DataRow y nos ubicamos en el 1er registro devuelto
    Dim DatosTabla As Data.DataRow
    If odt.Rows.Count = 0 Then
        Exit Sub
    Else

        f = odt.Rows.Count - 1
        DatosTabla = odt.Rows(f)
        Call Me.CrearListViewEstudiante()
        '-----
        ' Cargamos los datos de los estudiantnes devueltos
        '-----
        f = 0
        While f <= odt.Rows.Count - 1

            DatosTabla = odt.Rows(f) ' movernos por los regs resultantes
            Me.lvEstudiantes.Items.Add(DatosTabla("idEstudiante"))
            Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("Nombres"))
            Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("Apellidos"))
            f += 1
        End While
        'Cerramos o liberamos recursos
        odt.Dispose()
        oda.Dispose()
        ocmd.Dispose()
        cn.Close()
        cn.Dispose()
    End If
```



```
End If
Catch ex As Exception
    System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub

Sub CrearListViewAranceles()
    'Formatear el control ListView
    With Me.lvAranceles
        .Clear()

        .LabelEdit = False
        .AllowColumnReorder = True
        .FullRowSelect = True
        .Sorting = SortOrder.Ascending
        .View = View.Details
        .GridLines = True
        .Columns.Add("Id", 40, HorizontalAlignment.Center)
        .Columns.Add("Nombre", 150, HorizontalAlignment.Center)
        .Columns.Add("Costo", 150, HorizontalAlignment.Center)
        .Columns.Add("Descripción", 100, HorizontalAlignment.Center)

    End With
End Sub

Sub CargarAranceles()
    Dim CadenaConexion As String
    CadenaConexion = modUtilidades.Conexion
    Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
    Dim vListarArancel As New Data.SqlClient.SqlCommand("proc_ListarAranceles",
cn)

    vListarArancel.CommandType = CommandType.StoredProcedure
    Dim drListarAranceles As Data.SqlClient.SqlDataReader
    cn.Open()
    drListarAranceles = vListarArancel.ExecuteReader
    'Llenar el ListView
    Dim vpos As Integer
    Me.lvAranceles.Items.Clear()
    While drListarAranceles.Read
        With Me.lvAranceles
            .Items.Add(drListarAranceles("idArancel")) ' Identificador
            .Items(vpos).SubItems.Add(drListarAranceles("Nombre")) ' Nombre
            .Items(vpos).SubItems.Add(drListarAranceles("Costo")) ' Costo
            .Items(vpos).SubItems.Add(drListarAranceles("Descripcion")) ' Descripcion
        End With
    End While
End Sub
```



```
' Llenar el combo con los nombres de las aulas
Me.cboAranceles.Items.Add(Trim(drListarAranceles("Nombre"))) ' Nombre
vpos += 1
End With
End While
If Me.cboAranceles.Items.Count = 0 Then
    Me.cmdAceptarIdArancel.Enabled = False
    Me.cboAranceles.Enabled = False
    Me.cmdCancelarIdArancel.Focus()
Else
    Me.cmdAceptarIdArancel.Enabled = True
    Me.cmdCancelarIdArancel.Enabled = True
    Me.cboApellidosEstudiantes.Enabled = True

    Me.cboAranceles.SelectedIndex = 0
    Me.cmdAceptarIdArancel.Focus()
End If
'Cerrando objetos
drListarAranceles.Close()
cn.Close()
End Sub

Sub ArancelesXNombre()
    Dim CadenaConexion As String
    Dim f As Integer
    Try
        CadenaConexion = modUtilidades.Conexion
        Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
        Dim ocmd As SqlCommand = New
        SqlCommand("proc_ArancelesXNombre", cn)
        Dim oParametros As New SqlParameter()
        'nombre del aula a buscar
        If Me.cboAranceles.Text = "" Then
            Call Me.CargarAranceles()
        Else
            oParametros = ocmd.Parameters.Add("@nombre", SqlDbType.Char, 500)
            oParametros.Value = Me.cboAranceles.Text
            ocmd.CommandType = CommandType.StoredProcedure
            cn.Open()
            Dim oda As New SqlDataAdapter(ocmd)
            Dim odt As New DataTable()
            oda.Fill(odt)
            ' Creamos un objeto DataRow y nos ubicamos en el 1er registro devuelto
            Dim DatosTabla As DataRow
            If odt.Rows.Count = 0 Then
```



```
Exit Sub
Else
    f = odt.Rows.Count - 1
    DatosTabla = odt.Rows(f)
    Call Me.CrearListViewAranceles()
    '-----
    ' Cargamos los datos de las aulas devueltas
    '-----
    f = 0
    While f <= odt.Rows.Count - 1
        DatosTabla = odt.Rows(f) 'movernos por los regs resultantes
        Me.lvAranceles.Items.Add(DatosTabla("idArancel"))
        Me.lvAranceles.Items(f).SubItems.Add(DatosTabla("Nombre"))
        Me.lvAranceles.Items(f).SubItems.Add(DatosTabla("Costo"))
        Me.lvAranceles.Items(f).SubItems.Add(DatosTabla("Descripcion"))
        f += 1
    End While
    '-----
    'Cerramos o liberamos recursos
    odt.Dispose()
    oda.Dispose()
    ocmd.Dispose()
    cn.Close()
    cn.Dispose()
End If
End If
Catch ex As Exception
    System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try

End Sub

Private Sub cboApellidosEstudiantes_KeyUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles cboApellidosEstudiantes.KeyUp
    Call Me.EstudiantesXApellidos()
End Sub

Private Sub cmdAceptarIdEstudiante_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cmdAceptarIdEstudiante.Click
    Dim FilaActual As ListViewItem
    Try
        FilaActual = Me.lvEstudiantes.SelectedItems(0)
        Me.txtIdEstudiante.Text = FilaActual.SubItems(0).Text
```



```
Me.grpEstudiantes.Visible = False
Me.cmdAgregar.Visible = True
Me.cmdQuitar.Visible = True
End If
Catch ex As Exception
System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub

Private Sub cmdCancelarIdEstudiante_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdCancelarIdEstudiante.Click
Me.grpEstudiantes.Visible = False
Me.cmdAgregar.Visible = True
Me.cmdQuitar.Visible = True
End Sub

Private Sub cmdCancelar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdCancelar.Click
Me.Hide()
Me.Close()

End Sub

Private Sub cmdBuscarEstudiante_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdBuscarEstudiante.Click
Me.grpEstudiantes.Visible = True
Me.cmdAgregar.Visible = False
Me.cmdQuitar.Visible = False
Call Me.CrearListViewEstudiante()
Call Me.CargarEstudiantes()
Me.cboApellidosEstudiantes.Focus()
End Sub

Private Sub cboAranceles_KeyUp(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyEventArgs) Handles cboAranceles.KeyUp
Call Me.ArancelesXNombre()
End Sub

Private Sub cmdBuscarArancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdBuscarArancel.Click

Me.grpAranceles.Visible = True
Me.cmdAgregar.Visible = False
Me.cmdQuitar.Visible = False
Call Me.CrearListViewAranceles()
Call Me.CargarAranceles()
```



```
Me.cboAranceles.Focus()  
End Sub
```

```
Private Sub cmdAceptarIdArancel_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles cmdAceptarIdArancel.Click  
Dim FilaActual As ListViewItem  
Try  
FilaActual = Me.lvAranceles.SelectedItems(0)  
Me.txtIdArancel.Text = FilaActual.SubItems(0).Text  
Me.txtDescripcion.Text = FilaActual.SubItems(1).Text  
Me.txtSubTotal.Text = FilaActual.SubItems(2).Text  
Me.grpAranceles.Visible = False  
Me.cmdAgregar.Visible = True  
Me.cmdQuitar.Visible = True  
End If  
Catch ex As Exception  
System.Windows.Forms.MessageBox.Show(ex.Message.ToString)  
End Try  
  
End Sub
```

```
Private Sub cmdCancelarIdArancel_Click(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles cmdCancelarIdArancel.Click  
Me.grpAranceles.Visible = False  
Me.cmdAgregar.Visible = True  
Me.cmdQuitar.Visible = True  
End Sub
```

```
Private Sub lvEstudiantes_DoubleClick(ByVal sender As Object, ByVal e As  
System.EventArgs) Handles lvEstudiantes.DoubleClick  
Dim FilaActual As ListViewItem  
Try  
FilaActual = Me.lvEstudiantes.SelectedItems(0)  
Me.txtIdEstudiante.Text = FilaActual.SubItems(0).Text  
  
Me.grpEstudiantes.Visible = False  
Me.cmdAgregar.Visible = True  
Me.cmdQuitar.Visible = True  
Catch ex As Exception  
System.Windows.Forms.MessageBox.Show(ex.Message.ToString)  
End Try  
  
End Sub
```



```
Private Sub IvAranceles_DoubleClick(ByVal sender As Object, ByVal e As System.EventArgs) Handles IvAranceles.DoubleClick
```

```
    Dim FilaActual As ListViewItem
```

```
    Try
```

```
        FilaActual = Me.IvAranceles.SelectedItems(0)
```

```
        Me.txtIdArancel.Text = FilaActual.SubItems(0).Text
```

```
        Me.txtDescripcion.Text = FilaActual.SubItems(1).Text
```

```
        Me.txtSubTotal.Text = FilaActual.SubItems(2).Text
```

```
        Me.grpAranceles.Visible = False
```

```
        Me.cmdAgregar.Visible = True
```

```
        Me.cmdQuitar.Visible = True
```

```
    Catch ex As Exception
```

```
        System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
```

```
    End Try
```

```
End Sub
```

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

```
    ' Hacer algunas validaciones
```

```
    ' Agregar un detalle a la vista IvDetalleRecibo
```

```
    With Me.IvDetalleRecibo
```

```
        .Items.Add(Me.txtIdRecibo.Text)
```

```
        .Items(Fila).SubItems.Add(Me.txtIdArancel.Text)
```

```
        .Items(Fila).SubItems.Add(Me.txtDescripcion.Text)
```

```
        .Items(Fila).SubItems.Add(Me.txtAnyoPagado.Text)
```

```
        .Items(Fila).SubItems.Add(Me.txtMesPagado.Text)
```

```
        .Items(Fila).SubItems.Add(Me.txtSubTotal.Text)
```

```
        Fila += 1
```

```
    End With
```

```
    Me.txtTotal.Text = Val(Me.txtTotal.Text) + Val(Me.txtSubTotal.Text)
```

```
End Sub
```

```
Sub InsertarNuevoRecibo()
```

```
    Dim CadenaConexion As String
```

```
    Try
```

```
        CadenaConexion = modUtilidades.Conexion
```

```
        Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
```

```
        Dim cmd As SqlCommand = New
```

```
        SqlCommand("proc_InsertarRecibo", cn)
```

```
        Dim Parametros As New SqlParameter()
```

```
        'Identificador del recibo
```

```
        Parametros = cmd.Parameters.Add("@IdRecibo", SqlDbType.Char, 8)
```

```
        Parametros.Value = Me.txtIdRecibo.Text
```

```
        ' Identificador del estudiante
```



```
Parametros = cmd.Parameters.Add("@IdEstudiante", SqlDbType.Char, 8)
Parametros.Value = Me.txtIdEstudiante.Text
' Referencia del recibo
Parametros = cmd.Parameters.Add("@Referencia", SqlDbType.Char, 8)
Parametros.Value = Me.txtReferencia.Text
' Fecha del recibo
Parametros = cmd.Parameters.Add("@FechaRecibo", SqlDbType.DateTime)
Parametros.Value = Mid(Me.dtpFecha.Value, 1, 10) 'Me.dtpFecha.Value
' Concepto del recibo
Parametros = cmd.Parameters.Add("@Concepto", SqlDbType.Char, 1000)
Parametros.Value = Me.txtConcepto.Text
' Total
Parametros = cmd.Parameters.Add("@Total", SqlDbType.Money)
Parametros.Value = Me.txtTotal.Text
' Valor del recibo en letras
Parametros = cmd.Parameters.Add("@ValorEnLetras", SqlDbType.Char, 1000)
Parametros.Value = Me.txtValorEnLetras.Text
cmd.CommandType = CommandType.StoredProcedure
cn.Open()
Dim da As New Data.SqlClient.SqlDataAdapter(cmd)
Dim ds As New Data.DataSet()
da.Fill(ds)
' Cerramos o liberamos recursos
ds.Dispose()
da.Dispose()
cmd.Dispose()
cn.Close()
cn.Dispose()
Catch ex As Exception
    System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub

Sub InsertarNuevoDetalleRecibo(ByVal idRecibo As String, ByVal idArancel As
String, ByVal Descripcion As String, ByVal AnyoPagado As Integer, ByVal MesPagado
As Integer, ByVal SubTotal As Double)
    Dim CadenaConexion As String
    Try
        CadenaConexion = modUtilidades.Conexion
        Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
        Dim cmd As SqlCommand = New
        SqlCommand("proc_InsertarDetalleRecibo", cn)
        Dim Parametros As New SqlParameter()
        ' Identificador del recibo
```



```
Parametros = cmd.Parameters.Add("@IdRecibo", SqlDbType.Char, 8)
Parametros.Value = idRecibo
' Identificador del arancel
Parametros = cmd.Parameters.Add("@IdArancel", SqlDbType.Char, 4)
Parametros.Value = idArancel
' Descripcion
Parametros = cmd.Parameters.Add("@Descripcion", SqlDbType.Char, 500)
Parametros.Value = Descripcion
' Año pagado
Parametros = cmd.Parameters.Add("@AnyoPagado", SqlDbType.Int)
Parametros.Value = AnyoPagado
' Mes pagado
Parametros = cmd.Parameters.Add("@MesPagado", SqlDbType.Int)
Parametros.Value = MesPagado
' Sub Total
Parametros = cmd.Parameters.Add("@SubTotal", SqlDbType.Money)
Parametros.Value = SubTotal
cmd.CommandType = CommandType.StoredProcedure
cn.Open()
Dim da As New Data.SqlClient.SqlDataAdapter(cmd)
Dim ds As New Data.DataSet()
da.Fill(ds)
'Cerramos o liberamos recursos
ds.Dispose()
da.Dispose()
cmd.Dispose()
cn.Close()
cn.Dispose()
Catch ex As Exception
    System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub

Private Sub cmdQuitar_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles cmdQuitar.Click
    Try
        With Me.lvDetalleRecibo
            ' Evaluar que el usuario seleccione el detalle
            ' del recibo que desea eliminar
            If .SelectedItems.Count <> 0 Then
                .Items.Remove(.SelectedItems.Item(.SelectedItems.Count - 1))
                Fila -= 1
            Else
                MsgBox("Seleccione el detalle que desea quitar...")
            End If
        End With
    End Try
End Sub
```



```
        Me.cmdQuitar.Focus()
    Exit Sub
End If
' Activar o desactivar el Botón cmdBorrarDetalleTrabajo
' en caso necesario
If Fila > 0 Then
    Me.cmdQuitar.Enabled = True
Elseif Fila = 0 Then
    Me.cmdQuitar.Enabled = False
End If
End With
Catch ex As Exception
    System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub
```

```
Private Sub frmRecibo_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Me.txtTotal.Text = "0"
    Me.txtValorEnLetras.Text = "CERO"
    Me.txtAnyoPagado.Text = Mid(DateString, 7, 4)
    Me.txtMesPagado.Text = Mid(DateString, 1, 2)
    Call Me.CrearListViewDetalleR()
    Call Me.UltimoCodigoRecibo_Referencia()
End Sub
```

```
Private Sub cmdGuardar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdGuardar.Click
    Dim i As Integer
    Dim j As Integer
    Dim Filas1 As Integer
    Dim Columnas As Integer
    Dim IdReciboAux As String
    Dim IdArancelAux As String
    Dim DescripcionAux As String
    Dim AnyoPagadoAux As Integer
    Dim MesPagadoAux As Integer
    Dim SubTotalAux As Double
    Try
        ' Insertar el nuevo recibo
        Call Me.InsertarNuevoRecibo()
        ' Insertar el detalle del nuevo recibo
        Filas1 = 0
        Columnas = 0
```



```
With Me.lvDetalleRecibo
  Filas1 = .Items.Count - 1
  Columnas = .Columns.Count - 1
  For i = 0 To Filas1
    For j = 0 To Columnas
      IdReciboAux = .Items(i).SubItems.Item(0).Text
      IdArancelAux = .Items(i).SubItems.Item(1).Text
      DescripcionAux = .Items(i).SubItems.Item(2).Text
      AnyoPagadoAux = .Items(i).SubItems.Item(3).Text
      MesPagadoAux = .Items(i).SubItems.Item(4).Text
      SubTotalAux = .Items(i).SubItems.Item(5).Text
      Call Me.InsertarNuevoDetalleRecibo(IdReciboAux, IdArancelAux,
DescripcionAux, AnyoPagadoAux, MesPagadoAux, SubTotalAux)
    If j = 0 Then
      j = Columnas
    End If
  Next
Next
End With
MsgBox("Recibo insertado satisfactoriamente en el sistema")
Me.Hide()
Me.Close()
Catch ex As Exception
  System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub
End Class
```

frmAnularRecibo

```
Public Class frmAnularRecibo
  Sub CrearListViewRecibo()
    With Me.lvRecibos
      .LabelEdit = False
      .AllowColumnReorder = True
      .FullRowSelect = True
      .Sorting = SortOrder.Ascending
      .View = View.Details
      .GridLines = True
      .Columns.Add("Recibo", 65, HorizontalAlignment.Center)
      .Columns.Add("Estudiante", 70, HorizontalAlignment.Center)
      .Columns.Add("Referencia", 75, HorizontalAlignment.Center)
      .Columns.Add("Fecha", 80, HorizontalAlignment.Center)
      .Columns.Add("Concepto", 200, HorizontalAlignment.Center)
      .Columns.Add("Total", 70, HorizontalAlignment.Center)
    End With
  End Sub
End Class
```



```
.Columns.Add("Valor en letras", 100, HorizontalAlignment.Center)  
End With  
End Sub
```

```
Sub CrearListViewDetalleR()  
With Me.lvDetalleRecibo  
.Clear()  
.LabelEdit = False  
.AllowColumnReorder = True  
.FullRowSelect = True  
.Sorting = SortOrder.Ascending  
.View = View.Details  
.GridLines = True  
.Columns.Add("Id. Recibo", 75, HorizontalAlignment.Center)  
.Columns.Add("Id. Arancel", 75, HorizontalAlignment.Center)  
.Columns.Add("Descripción", 230, HorizontalAlignment.Center)  
.Columns.Add("Año pagado", 75, HorizontalAlignment.Center)  
.Columns.Add("Mes pagado", 75, HorizontalAlignment.Center)  
.Columns.Add("Sub Total", 80, HorizontalAlignment.Center)  
End With  
End Sub
```

```
Sub RecibosXFecha()  
Dim CadenaConexion As String  
Dim f As Integer  
Try  
CadenaConexion = modUtilidades.Conexion  
Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)  
Dim cmd As SqlCommand = New  
SqlCommand("proc_RptListarRecibosXFecha", cn)  
Dim Parametros As New SqlParameter()  
'Fecha  
Parametros = cmd.Parameters.Add("@Fecha", SqlDbType.DateTime)  
' Usamos mid con estos valores para q no tome en cuenta la  
' hora del sistema ya q el control pone la fecha y la hora  
Parametros.Value = Mid(Me.dtpFecha.Value, 1, 10)  
cmd.CommandType = CommandType.StoredProcedure  
cn.Open()  
Dim da As New Data.SqlClient.SqlDataAdapter(cmd)  
Dim dt As New DataTable()  
da.Fill(dt)  
' Creamos un objeto DataRow y nos ubicamos en el 1er registro devuelto  
Dim DatosTabla As DataRow  
If dt.Rows.Count = 0 Then
```



```
Call Me.CrearListViewRecibo()
Exit Sub
Else
Call Me.CrearListViewRecibo()
' Cargamos los datos de los recibos devueltos
f = 0
While f <= dt.Rows.Count - 1
    DatosTabla = dt.Rows(f) ' me muevo por los regs resultantes
    Me.lvRecibos.Items.Add(DatosTabla("idRecibo"))
    Me.lvRecibos.Items(f).SubItems.Add(DatosTabla("idEstudiante"))
    Me.lvRecibos.Items(f).SubItems.Add(DatosTabla("Referencia"))
    Me.lvRecibos.Items(f).SubItems.Add(DatosTabla("FechaRecibo"))
    Me.lvRecibos.Items(f).SubItems.Add(DatosTabla("Concepto"))
    Me.lvRecibos.Items(f).SubItems.Add(DatosTabla("Total"))
    Me.lvRecibos.Items(f).SubItems.Add(DatosTabla("ValorEnLetras"))
    f += 1
End While
'Cerramos o liberamos recursos
dt.Dispose()
da.Dispose()
cmd.Dispose()
cn.Close()
cn.Dispose()
End If
Catch ex As Exception
    System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub

Private Sub frmAnularRecibo_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Call Me.CrearListViewRecibo()
    Call Me.CrearListViewDetalleR()
End Sub

Private Sub cmdBuscarRecibos_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdBuscarRecibos.Click
    Call Me.RecibosXFecha()
End Sub

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



```
Me.Hide()  
Me.Close()  
End Sub
```

```
' Cuando se cambia el valor de la fecha, mandamos a llamar al procedimiento  
Private Sub mskFecha_ValueChanged(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles dtpFecha.ValueChanged  
    Call Me.RecibosXFecha()  
End Sub
```

```
Private Sub lvRecibos_DoubleClick(ByVal sender As Object, ByVal e As  
System.EventArgs) Handles lvRecibos.DoubleClick  
    Dim FilaActual As ListViewItem  
    Dim IdRecibo As String  
    Dim CadenaConexion As String  
    Dim f As Integer  
    Try  
        FilaActual = Me.lvRecibos.SelectedItems(0)  
        IdRecibo = FilaActual.SubItems(0).Text  
        CadenaConexion = modUtilidades.Conexion  
        Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)  
        Dim cmd As SqlCommand = New  
        SqlCommand("proc_ListarRecibosXIdRecibo", cn)  
        Dim Parametros As New SqlParameter()  
        'Identificador del recibo  
        Parametros = cmd.Parameters.Add("@IdRecibo", SqlDbType.Char, 8)  
        Parametros.Value = IdRecibo  
        cmd.CommandType = CommandType.StoredProcedure  
        cn.Open()  
        Dim da As New Data.SqlClient.SqlDataAdapter(cmd)  
        Dim dt As New DataTable()  
        da.Fill(dt)  
        'Creo un objeto DataRow y nos ubicamos en el 1er registro devuelto  
        Dim DatosTabla As DataRow  
        If dt.Rows.Count = 0 Then  
            Call Me.CrearListViewDetalleR()  
            MsgBox("Este recibo no tiene detalle ...", MsgBoxStyle.Exclamation, "ISSS")  
            Exit Sub  
        Else  
            Call Me.CrearListViewDetalleR()  
            '-----  
            ' Cargo los datos de los detalles del recibo devuelto  
            '-----
```



```
f = 0
While f <= dt.Rows.Count - 1
    DatosTabla = dt.Rows(f) ' me muevo por los regs resultantes
    Me.lvDetalleRecibo.Items.Add(DatosTabla("idRecibo"))
    Me.lvDetalleRecibo.Items(f).SubItems.Add(DatosTabla("idArancel"))
    Me.lvDetalleRecibo.Items(f).SubItems.Add(DatosTabla("Descripcion"))
    Me.lvDetalleRecibo.Items(f).SubItems.Add(DatosTabla("AnyoPagado"))
    Me.lvDetalleRecibo.Items(f).SubItems.Add(DatosTabla("MesPagado"))
    Me.lvDetalleRecibo.Items(f).SubItems.Add(DatosTabla("SubTotal"))
    f += 1
End While
'Cerramos o liberamos recursos
dt.Dispose()
da.Dispose()
cmd.Dispose()
cn.Close()
cn.Dispose()
End If
Catch ex As Exception
    System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub

Private Sub cmdAnular_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdAnular.Click
    Dim FilaActual As ListViewItem
    Dim IdRecibo As String
    Dim Concepto As String
    Dim CadenaConexion As String
    Dim FechaAux As Date
    ' Obtenemos la fecha del sistema
    FechaAux = Now()
    Try
        With Me.lvRecibos
            ' Evaluar q el usuario haya seleccionado un recibo
            If .SelectedItems.Count = 0 Then
                MsgBox("Debe seleccionar el recibo a anular", MsgBoxStyle.Exclamation)
                Exit Sub
            Else
                FilaActual = Me.lvRecibos.SelectedItems(0)
                ' Obtenemos el contenido del 1er elemento de la fila
                ' seleccionada por el usuario (el idRecibo)
                IdRecibo = FilaActual.SubItems(0).Text
                ' Obtenemos el contenido del 4to elemento de la fila
```



```
' seleccionada por el usuario (el Concepto)
Concepto = FilaActual.SubItems(4).Text
CadenaConexion = modUtilidades.Conexion
Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
Dim ocmd As SqlCommand = New
SqlCommand("proc_AnularRecibo", cn)
Dim oParametros As New SqlParameter()
'Identificador del recibo
oParametros = ocmd.Parameters.Add("@IdRecibo", SqlDbType.Char, 8)
oParametros.Value = IdRecibo
'Concepto del recibo
oParametros = ocmd.Parameters.Add("@Concepto", SqlDbType.Char,
1000)
oParametros.Value = Trim(FilaActual.SubItems(4).Text) & "." & " ANULADO
EL: " & FechaAux & "."
ocmd.CommandType = CommandType.StoredProcedure
cn.Open()
Dim oda As New Data.SqlClient.SqlDataAdapter(ocmd)
Dim ods As New DataSet()
oda.Fill(ods)
ods.Dispose()
oda.Dispose()
ocmd.Dispose()
cn.Close()
cn.Dispose()
MsgBox("Recibo anulado", MsgBoxStyle.Exclamation)
Call Me.RecibosXFecha()
Call Me.CrearListViewDetalleR()
End If
End With
Catch ex As Exception
System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub
End Class
```

frmCatAranceles

```
Public Class frmCatAranceles
Sub CrearListView()
With Me.lvAranceles
.Clear()
.LabelEdit = False
.AllowColumnReorder = True
```



```
.FullRowSelect = True
.Sorting = SortOrder.Ascending
.View = View.Details
.GridLines = True
' Aqui especificamos las columnas q va a tener el ListView
.Columns.Add("Id", 40, HorizontalAlignment.Center)
.Columns.Add("Nombre", 250, HorizontalAlignment.Center)
.Columns.Add("Costo", 100, HorizontalAlignment.Center)
.Columns.Add("Descripción", 250, HorizontalAlignment.Center)
```

```
End With
```

```
End Sub
```

```
Sub CargarAranceles()
```

```
Dim CadenaConexion As String
```

```
CadenaConexion = modUtilidades.Conexion
```

```
Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
```

```
Dim vListarArancel As New Data.SqlClient.SqlCommand("proc_ListarAranceles",
```

```
cn)
```

```
vListarArancel.CommandType = CommandType.StoredProcedure
```

```
Dim drListarAranceles As Data.SqlClient.SqlDataReader
```

```
cn.Open()
```

```
drListarAranceles = vListarArancel.ExecuteReader
```

```
'Llenar el ListView
```

```
Dim vpos As Integer
```

```
Me.lvAranceles.Items.Clear()
```

```
' Mientras el SqlDataReader tenga datos los agrego al listview
```

```
While drListarAranceles.Read
```

```
With Me.lvAranceles
```

```
' a drListarAranceles le pasamos como parametro el nombre
```

```
' del campo que quiero mostrar
```

```
.Items.Add(drListarAranceles("idArancel")) ' Identificador
```

```
.Items(vpos).SubItems.Add(drListarAranceles("Nombre")) ' Nombre
```

```
.Items(vpos).SubItems.Add(drListarAranceles("Costo")) ' Costo
```

```
.Items(vpos).SubItems.Add(drListarAranceles("Descripcion")) ' Descripcion
```

```
' Llenar el combo con los nombres de las aulas
```

```
Me.cboNombreAranceles.Items.Add(Trim(drListarAranceles("Nombre"))) '
```

```
Nombre
```

```
vpos += 1
```

```
End With
```

```
End While
```

```
' Verificamos si el combo tiene elementos
```

```
' también pudimos haber verificado con el
```

```
' control ListView
```

```
If Me.cboNombreAranceles.Items.Count = 0 Then
```



```
Me.cmdBorrar.Enabled = False
Me.cmdModificar.Enabled = False
Me.cboNombreAranceles.Enabled = False
Me.cmdNuevo.Focus()
Else
Me.cmdBorrar.Enabled = True
Me.cmdModificar.Enabled = True
Me.cboNombreAranceles.Enabled = True
Me.cboNombreAranceles.SelectedIndex = 0
Me.cmdNuevo.Focus()
End If
'Cerrando objetos
drListarAranceles.Close()
cn.Close()
End Sub

Sub ArancelesXNombre()
Dim CadenaConexion As String
Dim f As Integer
Try
CadenaConexion = modUtilidades.Conexion
Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
Dim ocmd As SqlCommand = New
SqlCommand.SqlCommand("proc_ArancelesXNombre", cn)
Dim oParametros As New SqlParameter()
'nombre del aula a buscar
If Me.cboNombreAranceles.Text = "" Then
Call Me.CargarAranceles()
Else
oParametros = ocmd.Parameters.Add("@nombre", SqlDbType.Char, 500)
oParametros.Value = Me.cboNombreAranceles.Text
ocmd.CommandType = CommandType.StoredProcedure
cn.Open()
Dim oda As New SqlDataAdapter(ocmd)
Dim odt As New DataTable()
oda.Fill(odt)
' Creo un objeto DataRow y nos ubicamos en el 1er registro devuelto
Dim DatosTabla As DataRow
If odt.Rows.Count = 0 Then
Exit Sub
Else
Call Me.CrearListView()
' Cargo los datos de las aulas devueltas
f = 0
' Mientras la f sea menor o igual al numero de
```



```
' elementos (filas) del objeto DataTable
While f <= odt.Rows.Count - 1
    ' Asignamos al objeto DatosTabla la fila del
    ' objeto DataTable q se encuentra en la posicion
    DatosTabla = odt.Rows(f)
    Me.lvAranceles.Items.Add(DatosTabla("idArancel"))
    Me.lvAranceles.Items(f).SubItems.Add(DatosTabla("Nombre"))
    Me.lvAranceles.Items(f).SubItems.Add(DatosTabla("Costo"))
    Me.lvAranceles.Items(f).SubItems.Add(DatosTabla("Descripcion"))
    f += 1
End While
'-----
'Cerramos o liberamos recursos
odt.Dispose()
oda.Dispose()
ocmd.Dispose()
cn.Close()
cn.Dispose()
End If
End If
Catch ex As Exception
    System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try

End Sub

Private Sub cmdCancelar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdCancelar.Click
    Me.Close()
End Sub

Private Sub cmdModificar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdModificar.Click
    Dim FilaActual As ListViewItem
    Dim IdAux As String
    Dim NombreAux As String
    Dim CostoAux As Double
    Dim DescripcionAux As String
    Try
        If Me.lvAranceles.SelectedItems.Count = 0 Then
            MsgBox("Debe especificar el arancel a modificar", MsgBoxStyle.Exclamation)
            Exit Sub
        Else
            FilaActual = Me.lvAranceles.SelectedItems(0)
```



```
IdAux = FilaActual.SubItems(0).Text
NombreAux = Trim(FilaActual.SubItems(1).Text)
CostoAux = Trim(FilaActual.SubItems(2).Text)
DescripcionAux = Trim(FilaActual.SubItems(3).Text)
With frmModificarArancel
    .txtIdentificador.Text = IdAux
    .txtNombre.Text = NombreAux
    .txtCosto.Text = CostoAux
    .txtDescripcion.Text = DescripcionAux
    Me.Hide()
    Me.Close()
    .Show()
End With
End If
Catch ex As Exception
    System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub

Sub EliminarArancel()
    Dim IdAux As String
    Dim CadenaConexion As String
    Dim FilaActual As ListViewItem
    Try
        'Obtener el Identificador del aula a borrar
        FilaActual = Me.lvAranceles.SelectedItems(0)
        IdAux = FilaActual.SubItems(0).Text
        CadenaConexion = modUtilidades.Conexion
        Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
        Dim ocmd As SqlCommand = New
        SqlCommand("proc_EliminarArancel", cn)
        Dim oParametros As New SqlParameter()
        'Identificador
        oParametros = ocmd.Parameters.Add("@idArancel", SqlDbType.Char, 4)
        oParametros.Value = IdAux
        ocmd.CommandType = CommandType.StoredProcedure
        cn.Open()
        Dim oda As New Data.SqlClient.SqlDataAdapter(ocmd)
        Dim ods As New DataSet()
        oda.Fill(ods)
        'Cerramos o liberamos recursos
        ods.Dispose()
        oda.Dispose()
```



```
ocmd.Dispose()  
cn.Close()  
cn.Dispose()  
MsgBox("Arancel eliminado")  
Catch ex As Exception
```

```
System.Windows.Forms.MessageBox.Show(ex.Message.ToString)  
End Try  
End Sub
```

```
Private Sub cmdBorrar_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles cmdBorrar.Click  
If Me.lvAranceles.SelectedItems.Count = 0 Then  
MsgBox("Debe especificar el arancel a borrar", MsgBoxStyle.Exclamation)  
Exit Sub  
Else  
Call Me.EliminarArancel()  
Call Me.CargarAranceles()  
End If  
End Sub
```

```
Private Sub cmdNuevo_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles cmdNuevo.Click  
Me.Hide()  
Me.Close()  
frmNuevoArancel.Show()  
End Sub
```

```
Private Sub frmCatAranceles_Load(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MyBase.Load  
Call Me.CrearListView()  
Call Me.CargarAranceles()  
End Sub
```

```
Private Sub cboNombreAranceles_KeyUp(ByVal sender As Object, ByVal e As  
System.Windows.Forms.KeyEventArgs) Handles cboNombreAranceles.KeyUp  
Call Me.ArancelesXNombre()  
End Sub  
End Class
```



frmNuevoArancel

```
Public Class frmNuevoArancel
    Sub UltimoCodigoArancel()
        Dim CadenaConexion As String

        CadenaConexion = modUtilidades.Conexion
        Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
        Dim vUltimoCodigo As New
Data.SqlClient.SqlCommand("proc_GenerarIdArancel", cn)
        vUltimoCodigo.CommandType = CommandType.StoredProcedure
        Dim drUltimoCodigo As Data.SqlClient.SqlDataReader
        cn.Open()
        drUltimoCodigo = vUltimoCodigo.ExecuteReader
        If drUltimoCodigo.Read Then
            ' Ubicamos en la caja de texto txtIdentificador el valor q retorno el procedimiento
            ' almacenado. Este valor lo ponemos formateado con 4 digitos
            Me.txtIdentificador.Text = Format(drUltimoCodigo("MaxCodigo"), "0000")
        Else
            Me.txtIdentificador.Text = Format("1", "0000")
        End If
        'Cerrando objetos
        drUltimoCodigo.Close()
        cn.Close()
    End Sub

    Sub InsertarNuevoArancel()
        Dim CadenaConexion As String
        Try
            CadenaConexion = modUtilidades.Conexion
            Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
            Dim cmd As SqlCommand = New
SqlClient.SqlCommand("proc_InsertarArancel", cn)
            Dim Parametros As New SqlParameter()
            'codigo
            Parametros = cmd.Parameters.Add("@id", SqlDbType.Char, 4)
            Parametros.Value = Me.txtIdentificador.Text
            'nombre
            Parametros = cmd.Parameters.Add("@nombre", SqlDbType.Char, 500)
            Parametros.Value = Trim(Me.txtNombre.Text)
            'costo
            Parametros = cmd.Parameters.Add("@costo", SqlDbType.Money)
            Parametros.Value = Trim(Me.txtCosto.Text)
            'descripcion
            Parametros = cmd.Parameters.Add("@descripcion", SqlDbType.Char, 500)
```



```
Parametros.Value = Trim(Me.txtDescripcion.Text)
cmd.CommandType = CommandType.StoredProcedure
cn.Open()
Dim da As New Data.SqlClient.SqlDataAdapter(cmd)
Dim ds As New Data.DataSet()
da.Fill(ds)
'Cerramos o liberamos recursos
ds.Dispose()
da.Dispose()
cmd.Dispose()
cn.Close()
cn.Dispose()
Catch ex As Exception
    System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub

Private Sub frmArancel_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Call Me.UltimoCodigoArancel()
    Me.txtNombre.Clear()
    Me.txtCosto.Clear()
    Me.txtDescripcion.Clear()
End Sub

Private Sub cmdCancelar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdCancelar.Click
    Me.Hide()
    Me.Close()
    frmPrincipal.IsMdiContainer = True 'Se utiliza para especificar que el FrmPadre es
un Mdi Padre
    frmCatAranceles.MdiParent = frmPrincipal 'Indica que es un formulario secundario
    frmCatAranceles.Show() 'Muestra el formulario
End Sub

Private Sub cmdAceptar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdAceptar.Click
    If Len(Me.txtNombre.Text) = 0 Then
        MsgBox("Debe especificar el nombre del arancel", MsgBoxStyle.Critical, "INSS")
        Me.txtNombre.Focus()
        Exit Sub
    ElseIf Len(Me.txtCosto.Text) = 0 Then
        MsgBox("Debe especificar el costo del arancel", MsgBoxStyle.Critical, "INSS")
```



```
Me.txtCosto.Focus()
Exit Sub
Elseif Len(Me.txtDescripcion.Text) = 0 Then
    MsgBox("Debe especificar la descripción del arancel", MsgBoxStyle.Critical,
    "INSS")
    Me.txtDescripcion.Focus()
    Exit Sub
Else
    'se llama antes de la insercion para que no haya una duplicidad
    ' en los datos(en el id)
    Call Me.UltimoCodigoArancel()
    Call Me.InsertarNuevoArancel()
    Me.Hide()
    Me.Close()
    frmPrincipal.IsMdiContainer = True 'Se utiliza para especificar que el FrmPadre
es un Mdi Padre
    frmCatAranceles.MdiParent = frmPrincipal 'Indica que es un formularios
ecundario
    frmCatAranceles.Show() 'Muestra el formulario
End If
End Sub
```

```
Private Sub txtCosto_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtCosto.KeyPress
    ' Solo permitimos numeros y el punto
    If (Asc(e.KeyChar)) >= 48 And (Asc(e.KeyChar)) <= 57 Or _
    (Asc(e.KeyChar)) = System.Windows.Forms.Keys.Back Or _
    (Asc(e.KeyChar)) = 46 Then
        e.Handled = False
    Else
        e.Handled = True
    End If
End Sub
```

```
Private Sub txtNombre_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtNombre.KeyPress
    ' Aqui permitimos solo letras, comas, puntos, espacios en blanco y borrar
    ' e.KeyChar contiene la tecla pulsada, entonces
    ' mediante la funcion Asc obtenemos el valor ASCII
    If (Asc(e.KeyChar)) >= 65 And (Asc(e.KeyChar)) <= 90 Or _
    (Asc(e.KeyChar)) >= 97 And (Asc(e.KeyChar)) <= 122 Or _
    (Asc(e.KeyChar)) = System.Windows.Forms.Keys.Back Or _
    (Asc(e.KeyChar)) = 44 Or (Asc(e.KeyChar)) = 46 Or _
    (Asc(e.KeyChar)) = 241 Or (Asc(e.KeyChar)) = 209 Or _
```



```
(Asc(e.KeyChar)) = 160 Or (Asc(e.KeyChar)) = 130 Or _  
(Asc(e.KeyChar)) = 161 Or (Asc(e.KeyChar)) = 162 Or _  
(Asc(e.KeyChar)) = 163 Or (Asc(e.KeyChar)) = 129 Or _  
(Asc(e.KeyChar)) = 154 Or _  
(Asc(e.KeyChar)) = 32 Then  
e.Handled = False  
Else e.Handled = True  
End If  
End Sub
```

```
Private Sub txtDescripcion_KeyPress(ByVal sender As Object, ByVal e As  
System.Windows.Forms.KeyPressEventArgs) Handles txtDescripcion.KeyPress  
If (Asc(e.KeyChar)) >= 65 And (Asc(e.KeyChar)) <= 90 Or _  
(Asc(e.KeyChar)) >= 97 And (Asc(e.KeyChar)) <= 122 Or _  
(Asc(e.KeyChar)) = System.Windows.Forms.Keys.Back Or _  
(Asc(e.KeyChar)) = 44 Or (Asc(e.KeyChar)) = 46 Or _  
(Asc(e.KeyChar)) = 241 Or (Asc(e.KeyChar)) = 209 Or _  
(Asc(e.KeyChar)) = 160 Or (Asc(e.KeyChar)) = 130 Or _  
(Asc(e.KeyChar)) = 161 Or (Asc(e.KeyChar)) = 162 Or _  
(Asc(e.KeyChar)) = 163 Or (Asc(e.KeyChar)) = 129 Or _  
(Asc(e.KeyChar)) = 154 Or _  
(Asc(e.KeyChar)) = 32 Then  
e.Handled = False  
Else  
e.Handled = True  
End If  
End Sub  
End Class
```

frmModificarArancel

```
Public Class frmModificarArancel  
Sub ModificarArancel()  
Dim CadenaConexion As String  
Try  
CadenaConexion = modUtilidades.Conexion  
Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)  
Dim ocmd As SqlCommand = New  
SqlCommand("proc_ModificarAranceles", cn)  
Dim oParametros As New SqlParameter()  
'Identificador  
oParametros = ocmd.Parameters.Add("@idArancel", SqlDbType.Char, 4)  
oParametros.Value = Me.txtIdentificador.Text  
'nombre
```



```
oParametros = ocmd.Parameters.Add("@nombre", SqlDbType.Char, 500)
oParametros.Value = Me.txtNombre.Text
'costo
oParametros = ocmd.Parameters.Add("@costo", SqlDbType.Money)
oParametros.Value = Me.txtCosto.Text
'descripcion
oParametros = ocmd.Parameters.Add("@descripcion", SqlDbType.Char, 500)
oParametros.Value = Me.txtDescripcion.Text
ocmd.CommandType = CommandType.StoredProcedure
cn.Open()
Dim oda As New Data.SqlClient.SqlDataAdapter(ocmd)
Dim ods As New Data.DataSet()
oda.Fill(ods)
'Cerramos o liberamos recursos
ods.Dispose()
oda.Dispose()
ocmd.Dispose()
cn.Close()
cn.Dispose()
MsgBox("Arancel modificado", MsgBoxStyle.Exclamation)
Catch ex As Exception
    System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub

Private Sub cmdCancelar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdCancelar.Click
    Me.Hide()
    Me.Close()
    frmPrincipal.IsMdiContainer = True 'Se utiliza para especificar que el FrmPadre es
un Mdi Padre
    frmCatAranceles.MdiParent = frmPrincipal 'Indica que es un formulario secundario
    frmCatAranceles.Show() 'Muestra el formulario
End Sub

Private Sub cmdAceptar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdAceptar.Click
    If Len(Me.txtNombre.Text) = 0 Then
        MsgBox("Debe especificar el nombre del arancel", MsgBoxStyle.Critical, "INSS")
        Me.txtNombre.Focus()
        Exit Sub
    ElseIf Len(Me.txtCosto.Text) = 0 Then
        MsgBox("Debe especificar el costo del arancel", MsgBoxStyle.Critical, "INSS")
```



```
Me.txtCosto.Focus()
Exit Sub
Elseif Len(Me.txtDescripcion.Text) = 0 Then
    MsgBox("Debe especificar la descripción del arancel", MsgBoxStyle.Critical,
"INSS")
    Me.txtDescripcion.Focus()
    Exit Sub
Else
    Call Me.ModificarArancel()
    Me.Hide()
    Me.Close()
    frmPrincipal.IsMdiContainer = True 'Se utiliza para especificar que el FrmPadre
es un Mdi Padre
    frmCatAranceles.MdiParent = frmPrincipal 'Indica que es un formulario
secundario
    frmCatAranceles.Show() 'Muestra el formulario
End If
End Sub
```

```
Private Sub txtNombre_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtNombre.KeyPress
    If (Asc(e.KeyChar)) >= 65 And (Asc(e.KeyChar)) <= 90 Or _
(Asc(e.KeyChar)) >= 97 And (Asc(e.KeyChar)) <= 122 Or _
(Asc(e.KeyChar)) = System.Windows.Forms.Keys.Back Or _
(Asc(e.KeyChar)) = 44 Or (Asc(e.KeyChar)) = 46 Or _
(Asc(e.KeyChar)) = 241 Or (Asc(e.KeyChar)) = 209 Or _
(Asc(e.KeyChar)) = 160 Or (Asc(e.KeyChar)) = 130 Or _
(Asc(e.KeyChar)) = 161 Or (Asc(e.KeyChar)) = 162 Or _
(Asc(e.KeyChar)) = 163 Or (Asc(e.KeyChar)) = 129 Or _
(Asc(e.KeyChar)) = 154 Or _
(Asc(e.KeyChar)) = 32 Then
    e.Handled = False
Else
    e.Handled = True
End If
End Sub
```

```
Private Sub txtDescripcion_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtDescripcion.KeyPress
    If (Asc(e.KeyChar)) >= 65 And (Asc(e.KeyChar)) <= 90 Or _

(Asc(e.KeyChar)) >= 97 And (Asc(e.KeyChar)) <= 122 Or _
(Asc(e.KeyChar)) = System.Windows.Forms.Keys.Back Or _
(Asc(e.KeyChar)) = 44 Or (Asc(e.KeyChar)) = 46 Or _
```



```
(Asc(e.KeyChar)) = 241 Or (Asc(e.KeyChar)) = 209 Or _  
(Asc(e.KeyChar)) = 160 Or (Asc(e.KeyChar)) = 130 Or _  
(Asc(e.KeyChar)) = 161 Or (Asc(e.KeyChar)) = 162 Or _  
(Asc(e.KeyChar)) = 163 Or (Asc(e.KeyChar)) = 129 Or _  
(Asc(e.KeyChar)) = 154 Or _  
(Asc(e.KeyChar)) = 32 Then  
    e.Handled = False
```

```
Else
```

```
    e.Handled = True
```

```
End If
```

```
End Sub
```

```
Private Sub txtCosto_KeyPress(ByVal sender As Object, ByVal e As  
System.Windows.Forms.KeyPressEventArgs) Handles txtCosto.KeyPress
```

```
    If (Asc(e.KeyChar)) >= 48 And (Asc(e.KeyChar)) <= 57 Or _  
        (Asc(e.KeyChar)) = System.Windows.Forms.Keys.Back Or _  
        (Asc(e.KeyChar)) = 46 Then  
        e.Handled = False
```

```
    Else
```

```
        e.Handled = True
```

```
    End If
```

```
End Sub
```

```
End Class
```

frmNuevaMatricula

```
Public Class frmNuevaMatricula
```

```
    Sub UltimoCodigoMatricula()
```

```
        Dim CadenaConexion As String
```

```
        CadenaConexion = modUtilidades.Conexion
```

```
        Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
```

```
        Dim vUltimoCodigo As New
```

```
Data.SqlClient.SqlCommand("proc_GenerarIdMatricula", cn)
```

```
        vUltimoCodigo.CommandType = CommandType.StoredProcedure
```

```
        Dim drUltimoCodigo As Data.SqlClient.SqlDataReader
```

```
        cn.Open()
```

```
        drUltimoCodigo = vUltimoCodigo.ExecuteReader
```

```
        If drUltimoCodigo.Read Then
```

```
            Me.txtIdMatricula.Text = Format(drUltimoCodigo("MaxCodigo"), "00000000")
```

```
        Else
```

```
            Me.txtIdMatricula.Text = Format("1", "00000000")
```

```
        End If
```

```
        'Cerrando objetos
```

```
        drUltimoCodigo.Close()
```

```
        cn.Close()
```



End Sub

Sub InsertarNuevaMatricula()

Dim CadenaConexion As String

Try

CadenaConexion = modUtilidades.Conexion

Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)

Dim cmd As SqlCommand = New

SqlCommand("proc_InsertarMatricula", cn)

Dim Parametros As New SqlParameter()

'Identificador de la matricula

Parametros = cmd.Parameters.Add("@IdMatricula", SqlDbType.Char, 8)

Parametros.Value = Me.txtIdMatricula.Text

'Identificador del estudiante

Parametros = cmd.Parameters.Add("@IdEstudiante", SqlDbType.Char, 8)

Parametros.Value = Me.txtIdEstudiante.Text

'Identificador del aula

Parametros = cmd.Parameters.Add("@IdAula", SqlDbType.Char, 4)

Parametros.Value = Me.txtIdAula.Text

'Turno

Parametros = cmd.Parameters.Add("@Turno", SqlDbType.Char, 10)

Parametros.Value = Trim(Me.cboTurno.Text)

'Año lectivo

Parametros = cmd.Parameters.Add("@AnyoLectivo", SqlDbType.Int)

Parametros.Value = Me.txtAnyoLectivo.Text

'Grado matrícula

Parametros = cmd.Parameters.Add("@GradoMatricula", SqlDbType.Char, 5)

Parametros.Value = Trim(Me.cboGradoMatricula.Text)

'Fecha matrícula

Parametros = cmd.Parameters.Add("@FechaMatricula", SqlDbType.DateTime)

Parametros.Value = Mid(Me.dtpFechaMatricula.Value, 1, 10)

'Partida de nacimiento

Parametros = cmd.Parameters.Add("@PartidaNacimiento", SqlDbType.Char, 2)

Parametros.Value = Trim(Me.cboPartidaNac.Text)

'Notas del último grado

Parametros = cmd.Parameters.Add("@NotasUltimoGrado", SqlDbType.Char, 2)

Parametros.Value = Trim(Me.cboNotasUltimoGrado.Text)

'Diploma de sexto grado

Parametros = cmd.Parameters.Add("@DiplomaSextoGrado", SqlDbType.Char,

2)

Parametros.Value = Trim(Me.cboDipSextoGrado.Text)

'Otros documentos



```
Parametros = cmd.Parameters.Add("@OtrosDocumentos", SqlDbType.Char, 500)
Parametros.Value = Trim(Me.txtOtrosDoc.Text)
'Repitente

Parametros = cmd.Parameters.Add("@Repitente", SqlDbType.Char, 2)
Parametros.Value = Trim(Me.cboRepitente.Text)
'Observación
Parametros = cmd.Parameters.Add("@Observacion", SqlDbType.Char, 1000)
Parametros.Value = Trim(Me.txtObservacion.Text)
cmd.CommandType = CommandType.StoredProcedure
cn.Open()
Dim da As New Data.SqlClient.SqlDataAdapter(cmd)
Dim ds As New Data.DataSet()
da.Fill(ds)
'Cerramos o liberamos recursos
ds.Dispose()
da.Dispose()
cmd.Dispose()
cn.Close()
cn.Dispose()
MsgBox("Estudiante matriculado", MsgBoxStyle.Exclamation)
Catch ex As Exception
System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub

Sub CrearListViewAula()
'Formatear el control ListView
With Me.lvAulas
.Clear()
.LabelEdit = False
.AllowColumnReorder = True
.FullRowSelect = True
.Sorting = SortOrder.Ascending
.View = View.Details
.GridLines = True
.Columns.Add("Id", 40, HorizontalAlignment.Center)
.Columns.Add("Nombre", 200, HorizontalAlignment.Center)
.Columns.Add("Descripción", 250, HorizontalAlignment.Center)
End With
End Sub

Sub CargarAulas()
Dim CadenaConexion As String
```



```
CadenaConexion = modUtilidades.Conexion
Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
Dim vListarCargos As New Data.SqlClient.SqlCommand("proc_ListarAulas", cn)
vListarCargos.CommandType = CommandType.StoredProcedure

Dim drListarCargos As Data.SqlClient.SqlDataReader
cn.Open()
drListarCargos = vListarCargos.ExecuteReader
'Llenar el ListView
Dim vpos As Integer
Me.lvAulas.Items.Clear()
While drListarCargos.Read
    With Me.lvAulas
        .Items.Add(drListarCargos("idAula")) ' Identificador
        .Items(vpos).SubItems.Add(drListarCargos("Nombre")) ' Nombre
        .Items(vpos).SubItems.Add(drListarCargos("Descripcion")) ' Descripcion
        ' Llenar el combo con los nombres de las aulas
        Me.cboNombreAulas.Items.Add(Trim(drListarCargos("Nombre"))) ' Nombre
        vpos += 1
    End With
End While
If Me.cboNombreAulas.Items.Count = 0 Then
    Me.cmdPasarIdAula.Enabled = False
    Me.cboNombreAulas.Enabled = False
    Me.cmdPasarIdAula.Focus()
Else
    Me.cmdPasarIdAula.Enabled = True
    Me.cmdCancelarPasarIdAula.Enabled = True
    Me.cboNombreAulas.Enabled = True
    Me.cboNombreAulas.SelectedIndex = 0
    Me.cmdPasarIdAula.Focus()
End If
drListarCargos.Close()
cn.Close()
End Sub

Sub AulasXNombre()
    Dim CadenaConexion As String
    Dim f As Integer
    Try
        CadenaConexion = modUtilidades.Conexion
        Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
        Dim ocmd As SqlCommand = New
        SqlCommand("proc_AulasXNombre", cn)
```



```
Dim oParametros As New SqlClient.SqlParameter()  
'nombre del aula a buscar  
If Me.cboNombreAulas.Text = "" Then  
    Call Me.CargarAulas()  
Else  
  
    oParametros = ocmd.Parameters.Add("@nombre", SqlDbType.Char, 250)  
    oParametros.Value = Me.cboNombreAulas.Text  
    ocmd.CommandType = CommandType.StoredProcedure  
    cn.Open()  
    Dim oda As New Data.SqlClient.SqlDataAdapter(ocmd)  
    Dim odt As New DataTable()  
    oda.Fill(odt)  
    ' Creamos un objeto DataRow y nos ubicamos en el 1er registro devuelto  
    Dim DatosTabla As Data.DataRow  
    If odt.Rows.Count = 0 Then  
        Exit Sub  
    Else  
        f = odt.Rows.Count - 1  
        DatosTabla = odt.Rows(f)  
        Call Me.CrearListViewAula()  
        ' Cargo los datos de las aulas devueltas  
        f = 0  
        While f <= odt.Rows.Count - 1  
            DatosTabla = odt.Rows(f) ' me muevo por los regs resultantes  
            Me.lvAulas.Items.Add(DatosTabla("idAula"))  
            Me.lvAulas.Items(f).SubItems.Add(DatosTabla("Nombre"))  
            Me.lvAulas.Items(f).SubItems.Add(DatosTabla("Descripcion"))  
            f += 1  
        End While  
        odt.Dispose()  
        oda.Dispose()  
        ocmd.Dispose()  
        cn.Close()  
        cn.Dispose()  
    End If  
End If  
Catch ex As Exception  
    System.Windows.Forms.MessageBox.Show(ex.Message.ToString)  
End Try  
End Sub  
  
Sub CrearListView()  
    'Formatear el control ListView
```



With Me.lvEstudiantes

```
.Clear()  
.LabelEdit = False  
.AllowColumnReorder = True  
.FullRowSelect = True  
.Sorting = SortOrder.Ascending
```

```
.View = View.Details  
.GridLines = True  
.Columns.Add("Id", 70, HorizontalAlignment.Center)  
.Columns.Add("Nombres", 150, HorizontalAlignment.Center)  
.Columns.Add("Apellidos", 150, HorizontalAlignment.Center)  
.Columns.Add("Lugar Nac.", 100, HorizontalAlignment.Center)  
.Columns.Add("Fecha Nac.", 100, HorizontalAlignment.Center)  
.Columns.Add("Edad", 50, HorizontalAlignment.Center)  
.Columns.Add("Sexo", 50, HorizontalAlignment.Center)  
.Columns.Add("Tipo Sangre", 80, HorizontalAlignment.Center)  
.Columns.Add("Dirección", 200, HorizontalAlignment.Center)  
.Columns.Add("Teléfono", 100, HorizontalAlignment.Center)
```

End With

End Sub

Sub CargarEstudiantes()

Dim CadenaConexion As String

CadenaConexion = modUtilidades.Conexion

Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)

Dim vListarCargos As New Data.SqlClient.SqlCommand("proc_ListarEstudiantes",

cn)

vListarCargos.CommandType = CommandType.StoredProcedure

Dim drListarCargos As Data.SqlClient.SqlDataReader

cn.Open()

drListarCargos = vListarCargos.ExecuteReader

Llenar el ListView

Dim vpos As Integer

Me.lvEstudiantes.Items.Clear()

While drListarCargos.Read

With Me.lvEstudiantes

.Items.Add(drListarCargos("idEstudiante")) ' Identificador

.Items(vpos).SubItems.Add(drListarCargos("Nombres")) ' Nombres

.Items(vpos).SubItems.Add(drListarCargos("Apellidos")) ' Apellidos

.Items(vpos).SubItems.Add(drListarCargos("LugarNacimiento")) ' Lugar de

nacimiento

.Items(vpos).SubItems.Add(drListarCargos("FechaNacimiento")) ' Fecha de

nacimiento



```
.Items(vpos).SubItems.Add(drListarCargos("Edad")) ' Edad  
.Items(vpos).SubItems.Add(drListarCargos("Sexo")) ' Sexo  
.Items(vpos).SubItems.Add(drListarCargos("TipoSangre")) ' Tipo de sangre  
.Items(vpos).SubItems.Add(drListarCargos("Direccion")) ' Dirección  
.Items(vpos).SubItems.Add(drListarCargos("Telefono")) ' Teléfono  
' Llenar el combo con los nombres de las aulas
```

```
Me.cboApellidosEstudiantes.Items.Add(Trim(drListarCargos("Apellidos"))) '
```

Apellidos

```
vpos += 1
```

```
End With
```

```
End While
```

```
If Me.cboApellidosEstudiantes.Items.Count = 0 Then
```

```
Me.cmdAceptar.Enabled = False
```

```
Me.cboApellidosEstudiantes.Enabled = False
```

```
Me.cmdCancelar.Focus()
```

```
Else
```

```
Me.cmdAceptar.Enabled = True
```

```
Me.cmdCancelar.Enabled = True
```

```
Me.cboApellidosEstudiantes.Enabled = True
```

```
Me.cboApellidosEstudiantes.SelectedIndex = 0
```

```
Me.cmdAceptar.Focus()
```

```
End If
```

```
drListarCargos.Close()
```

```
cn.Close()
```

```
End Sub
```

```
Sub EstudiantesXApellidos()
```

```
Dim CadenaConexion As String
```

```
Dim f As Integer
```

```
Try
```

```
CadenaConexion = modUtilidades.Conexion
```

```
Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
```

```
Dim ocmd As SqlCommand = New
```

```
SqlCommand("proc_EstudiantesXApellidos", cn)
```

```
Dim oParametros As New SqlParameter()
```

```
'Apellidos del estudiante a buscar
```

```
If Me.cboApellidosEstudiantes.Text = "" Then
```

```
Call Me.CargarEstudiantes()
```

```
Else
```

```
oParametros = ocmd.Parameters.Add("@apellidos", SqlDbType.Char, 250)
```

```
oParametros.Value = Me.cboApellidosEstudiantes.Text
```

```
ocmd.CommandType = CommandType.StoredProcedure
```

```
cn.Open()
```



```
Dim oda As New Data.SqlClient.SqlDataAdapter(ocmd)
Dim odt As New DataTable()
oda.Fill(odt)
' Creo un objeto DataRow y nos ubicamos en el 1er registro devuelto
Dim DatosTabla As Data.DataRow
If odt.Rows.Count = 0 Then
    Exit Sub
Else
    f = odt.Rows.Count - 1
    DatosTabla = odt.Rows(f)
    Call Me.CrearListView()
    f = 0
    While f <= odt.Rows.Count - 1
        DatosTabla = odt.Rows(f) ' me muevo por los regs resultantes
        Me.lvEstudiantes.Items.Add(DatosTabla("idEstudiante"))
        Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("Nombres"))
        Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("Apellidos"))
        Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("LugarNacimiento"))
        Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("FechaNacimiento"))
        Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("Edad"))
        Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("Sexo"))
        Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("TipoSangre"))
        Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("Direccion"))
        Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("Telefono"))
        f += 1
    End While
    odt.Dispose()
    oda.Dispose()
    ocmd.Dispose()
    cn.Close()
    cn.Dispose()
End If
End If
Catch ex As Exception
    System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub

Sub LlenarCombos()
    With Me.cboTurno
        .Items.Add("Matutino")
        .Items.Add("Vespertino")
        .SelectedIndex = 0
    End With
```



```
With Me.cboGradoMatricula
```

```
.Items.Add("I")  
.Items.Add("II")  
.Items.Add("III")  
.Items.Add("IV")  
.Items.Add("V")  
.SelectedIndex = 0
```

```
End With
```

```
With Me.cboRepitente
```

```
.Items.Add("Si")  
.Items.Add("No")  
.SelectedIndex = 0
```

```
End With
```

```
With Me.cboPartidaNac
```

```
.Items.Add("Si")  
.Items.Add("No")  
.SelectedIndex = 0
```

```
End With
```

```
With Me.cboNotasUltimoGrado
```

```
.Items.Add("Si")  
.Items.Add("No")  
.SelectedIndex = 0
```

```
End With
```

```
With Me.cboDipSextoGrado
```

```
.Items.Add("Si")  
.Items.Add("No")  
.SelectedIndex = 0
```

```
End With
```

```
End Sub
```

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

```
Me.Hide()  
Me.Close()
```

```
End Sub
```

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

```
Call Me.UltimoCodigoMatricula()  
Call Me.LlenarCombos()  
Me.txtAnyoLectivo.Text = Mid(Me.dtpFechaMatricula.Value, 7, 4)  
Me.txtOtrosDoc.Text = "Ninguno"  
Me.txtObservacion.Text = "Ninguna"
```

```
End Sub
```



```
Private Sub cmdCancelarPasarIdEstu_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdCancelarPasarIdEstu.Click
    Me.grpEstudiantes.Visible = False
End Sub
```

```
Private Sub cmdPasarIdEstudiante_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdPasarIdEstudiante.Click
    Dim FilaActual As ListViewItem
    Try
        FilaActual = Me.lvEstudiantes.SelectedItems(0)
        Me.txtIdEstudiante.Text = FilaActual.SubItems(0).Text
        Me.grpEstudiantes.Visible = False
    Catch ex As Exception
        System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
    End Try
End Sub
```

```
Private Sub cmdCancelarPasarIdAula_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdCancelarPasarIdAula.Click
    Me.grpAulas.Visible = False
End Sub
```

```
Private Sub cboApellidosEstudiantes_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cboApellidosEstudiantes.SelectedIndexChanged
    Call Me.EstudiantesXApellidos()
End Sub
```

```
Private Sub cboNombreAulas_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cboNombreAulas.SelectedIndexChanged
    Call Me.AulasXNombre()
End Sub
```

```
Private Sub cmdBuscarEstudiantes_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdBuscarEstudiantes.Click
    Me.grpEstudiantes.Visible = True
    Me.grpAulas.Visible = False
    Call Me.CrearListView()
    Call Me.CargarEstudiantes()
    Me.cboApellidosEstudiantes.Focus()
End Sub
```



```
Private Sub cmdBuscarAulas_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdBuscarAulas.Click
    Me.grpAulas.Visible = True
    Me.grpEstudiantes.Visible = False
    Call Me.CrearListViewAula()
    Call Me.CargarAulas()
    Me.cboNombreAulas.Focus()
End Sub
```

```
Private Sub lvEstudiantes_DoubleClick(ByVal sender As Object, ByVal e As System.EventArgs) Handles lvEstudiantes.DoubleClick
    Dim FilaActual As ListViewItem
    Try
        FilaActual = Me.lvEstudiantes.SelectedItems(0)
        Me.txtIdEstudiante.Text = FilaActual.SubItems(0).Text
        Me.grpEstudiantes.Visible = False
    Catch ex As Exception
        System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
    End Try
End Sub
```

```
Private Sub lvAulas_DoubleClick(ByVal sender As Object, ByVal e As System.EventArgs) Handles lvAulas.DoubleClick
    Dim FilaActual As ListViewItem
    Try
        FilaActual = Me.lvAulas.SelectedItems(0)
        Me.txtIdAula.Text = FilaActual.SubItems(0).Text
        Me.grpAulas.Visible = False
    Catch ex As Exception
        System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
    End Try
End Sub
```

```
Private Sub cmdAceptar_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles cmdAceptar.Click
    If Len(Me.txtIdEstudiante.Text) = 0 Then
        MsgBox("Debe especificar el identificador del estudiante", MsgBoxStyle.Critical, "INSS")
        Me.cmdPasarIdEstudiante.Focus()
    Exit Sub
    ElseIf Len(Me.txtIdAula.Text) = 0 Then
        MsgBox("Debe especificar el identificador del aula", MsgBoxStyle.Critical, "INSS")
        Me.cmdPasarIdAula.Focus()
    End If
End Sub
```



```
Exit Sub
Elseif Len(Me.cboTurno.Text) = 0 Then
    MsgBox("Debe especificar el turno de la matrícula", MsgBoxStyle.Critical,
"INSS")
    Me.cboTurno.Focus()

Exit Sub
Elseif Len(Me.txtAnyoLectivo.Text) = 0 Then
    MsgBox("Debe especificar el año lectivo", MsgBoxStyle.Critical, "INSS")
    Me.txtAnyoLectivo.Focus()
Exit Sub
Elseif Len(Me.cboGradoMatricula.Text) = 0 Then
    MsgBox("Debe especificar el grado de la matrícula", MsgBoxStyle.Critical,
"INSS")
    Me.cboGradoMatricula.Focus()
Exit Sub
Elseif Len(Me.cboRepitente.Text) = 0 Then
    MsgBox("Debe especificar si es repitente o no", MsgBoxStyle.Critical, "INSS")
    Me.cboRepitente.Focus()
Exit Sub
Elseif Len(Me.cboPartidaNac.Text) = 0 Then
    MsgBox("Debe especificar la partida de nacimiento", MsgBoxStyle.Critical,
"INSS")
    Me.cboPartidaNac.Focus()
Exit Sub
Elseif Len(Me.cboNotasUltimoGrado.Text) = 0 Then
    MsgBox("Debe especificar notas del último año", MsgBoxStyle.Critical, "INSS")
    Me.cboNotasUltimoGrado.Focus()
Exit Sub
Elseif Len(Me.cboDipSextoGrado.Text) = 0 Then
    MsgBox("Debe especificar si trajo el diploma de sexto grado",
MsgBoxStyle.Critical, "INSS")
    Me.cboDipSextoGrado.Focus()
Exit Sub
Else
    Call Me.UltimoCodigoMatricula()
    Call Me.InsertarNuevaMatricula()
    Call Me.UltimoCodigoMatricula()
    Me.txtIdEstudiante.Text = ""
    Me.txtIdAula.Text = ""
    Me.txtOtrosDoc.Text = "Ninguno"
    Me.txtObservacion.Text = "Ninguna"
End If
```



End Sub

```
Private Sub txtAnyoLectivo_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtAnyoLectivo.KeyPress
```

```
    If (Asc(e.KeyChar)) >= 48 And (Asc(e.KeyChar)) <= 57 Or _  
        (Asc(e.KeyChar)) = System.Windows.Forms.Keys.Back Then
```

```
        e.Handled = False
```

```
    Else
```

```
        e.Handled = True
```

```
    End If
```

```
End Sub
```

```
Private Sub txtOtrosDoc_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtOtrosDoc.KeyPress
```

```
    If (Asc(e.KeyChar)) >= 65 And (Asc(e.KeyChar)) <= 90 Or _  
        (Asc(e.KeyChar)) >= 97 And (Asc(e.KeyChar)) <= 122 Or _  
        (Asc(e.KeyChar)) = System.Windows.Forms.Keys.Back Or _
```

```
        (Asc(e.KeyChar)) = 44 Or (Asc(e.KeyChar)) = 46 Or _
```

```
        (Asc(e.KeyChar)) = 241 Or (Asc(e.KeyChar)) = 209 Or _
```

```
        (Asc(e.KeyChar)) = 160 Or (Asc(e.KeyChar)) = 130 Or _
```

```
        (Asc(e.KeyChar)) = 161 Or (Asc(e.KeyChar)) = 162 Or _
```

```
        (Asc(e.KeyChar)) = 163 Or (Asc(e.KeyChar)) = 129 Or _
```

```
        (Asc(e.KeyChar)) = 154 Or _
```

```
        (Asc(e.KeyChar)) = 32 Then
```

```
        e.Handled = False
```

```
    Else
```

```
        e.Handled = True
```

```
    End If
```

```
End Sub
```

```
Private Sub txtObservacion_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtObservacion.KeyPress
```

```
    If (Asc(e.KeyChar)) >= 65 And (Asc(e.KeyChar)) <= 90 Or _
```

```
        (Asc(e.KeyChar)) >= 97 And (Asc(e.KeyChar)) <= 122 Or _
```

```
        (Asc(e.KeyChar)) = System.Windows.Forms.Keys.Back Or _
```

```
        (Asc(e.KeyChar)) = 44 Or (Asc(e.KeyChar)) = 46 Or _
```

```
        (Asc(e.KeyChar)) = 241 Or (Asc(e.KeyChar)) = 209 Or _
```

```
        (Asc(e.KeyChar)) = 160 Or (Asc(e.KeyChar)) = 130 Or _
```

```
        (Asc(e.KeyChar)) = 161 Or (Asc(e.KeyChar)) = 162 Or _
```

```
        (Asc(e.KeyChar)) = 163 Or (Asc(e.KeyChar)) = 129 Or _
```

```
        (Asc(e.KeyChar)) = 154 Or _
```

```
        (Asc(e.KeyChar)) = 32 Then
```

```
        e.Handled = False
```



```
Else
    e.Handled = True
End If
End Sub

Private Sub cmdPasarIdAula_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdPasarIdAula.Click
    Dim FilaActual As ListViewItem
    Try
        FilaActual = Me.lvAulas.SelectedItems(0)
        Me.txtIdAula.Text = FilaActual.SubItems(0).Text
        Me.grpAulas.Visible = False
    Catch ex As Exception
        System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
    End Try
End Sub
End Class
```

frmMatriculasXFecha

```
Public Class frmFechasMatriculas
    Sub CrearListView()
        'Formatear el control ListView
        With Me.lvMatriculas
            .Clear()
            .LabelEdit = False
            .AllowColumnReorder = True
            .FullRowSelect = True
            .Sorting = SortOrder.Ascending
            .View = View.Details
            .GridLines = True
            .Columns.Add("Id. Matrícula", 80, HorizontalAlignment.Center)
            .Columns.Add("Id. Estudiante", 80, HorizontalAlignment.Center)
            .Columns.Add("Id. Aula", 80, HorizontalAlignment.Center)
            .Columns.Add("Turno", 70, HorizontalAlignment.Center)
            .Columns.Add("Año Lectivo", 70, HorizontalAlignment.Center)
            .Columns.Add("Grado Matrícula", 100, HorizontalAlignment.Center)
            .Columns.Add("Fecha Matrícula", 100, HorizontalAlignment.Center)
            .Columns.Add("Partida Nac.", 100, HorizontalAlignment.Center)
            .Columns.Add("Notas Último Grado", 70, HorizontalAlignment.Center)
            .Columns.Add("Dip. 6to. Grado", 70, HorizontalAlignment.Center)
            .Columns.Add("Otros Doc.", 70, HorizontalAlignment.Center)
            .Columns.Add("Repitente", 70, HorizontalAlignment.Center)
            .Columns.Add("Observación", 100, HorizontalAlignment.Center)
        End With
    End Sub
End Class
```



End With
End Sub

Sub MatriculasXFecha()

Dim CadenaConexion As String

Dim f As Integer

Try

CadenaConexion = modUtilidades.Conexion

Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)

Dim ocmd As SqlCommand = New

SqlCommand("proc_RptListarMatriculasXFecha", cn)

Dim oParametros As New SqlParameter()

'Fecha inicial

oParametros = ocmd.Parameters.Add("@FechaInicial", SqlDbType.DateTime)

oParametros.Value = Mid(Me.dtpFechaInicial.Value, 1, 10)

'Fecha final

oParametros = ocmd.Parameters.Add("@FechaFinal", SqlDbType.DateTime)

oParametros.Value = Mid(Me.dtpFechaFinal.Value, 1, 10)

ocmd.CommandType = CommandType.StoredProcedure

cn.Open()

Dim oda As New Data.SqlClient.SqlDataAdapter(ocmd)

Dim odt As New DataTable()

oda.Fill(odt)

' Creamos un objeto DataRow y nos ubicamos en el 1er registro devuelto

Dim DatosTabla As DataRow

If odt.Rows.Count = 0 Then

Call Me.CrearListView()

Exit Sub

Else

f = odt.Rows.Count - 1

DatosTabla = odt.Rows(f)

Call Me.CrearListView()

' Cargamos los datos de las matrículas devueltas

f = 0

While f <= odt.Rows.Count - 1

DatosTabla = odt.Rows(f) ' nos movemos por los regs resultantes

Me.lvMatriculas.Items.Add(DatosTabla("idMatricula"))

Me.lvMatriculas.Items(f).SubItems.Add(DatosTabla("idEstudiante"))

Me.lvMatriculas.Items(f).SubItems.Add(DatosTabla("idAula"))

Me.lvMatriculas.Items(f).SubItems.Add(DatosTabla("Turno"))

Me.lvMatriculas.Items(f).SubItems.Add(DatosTabla("AnyoLectivo"))

Me.lvMatriculas.Items(f).SubItems.Add(DatosTabla("GradoMatricula"))



```
Me.lvMatriculas.Items(f).SubItems.Add(DatosTabla("FechaMatricula"))
Me.lvMatriculas.Items(f).SubItems.Add(DatosTabla("PartidaNacimiento"))
Me.lvMatriculas.Items(f).SubItems.Add(DatosTabla("NotasUltimoGrado"))
Me.lvMatriculas.Items(f).SubItems.Add(DatosTabla("DiplomaSextoGrado"))
Me.lvMatriculas.Items(f).SubItems.Add(DatosTabla("OtrosDocumentos"))
Me.lvMatriculas.Items(f).SubItems.Add(DatosTabla("Repitente"))
Me.lvMatriculas.Items(f).SubItems.Add(DatosTabla("Observacion"))
```

```
f += 1
```

```
End While
```

```
-----
```

```
odt.Dispose()
oda.Dispose()
ocmd.Dispose()
cn.Close()
cn.Dispose()
```

```
End If
```

```
Catch ex As Exception
```

```
System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
```

```
End Try
```

```
End Sub
```

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

```
Me.Hide()
```

```
Me.Close()
```

```
End Sub
```

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

```
Call Me.MatriculasXFecha()
```

```
End Sub
```

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

```
Call Me.CrearListView()
```

```
End Sub
```

```
Sub AnularRetirarMatricula()
```

```
Dim CadenaConexion As String
```

```
Dim FilaActual As ListViewItem
```

```
Dim FechaAux As Date
```

```
FechaAux = Now()
```

```
Try
```



```
'Obtener el Identificador de la matrícula a retirar
FilaActual = Me.lvMatriculas.SelectedItems(0)
CadenaConexion = modUtilidades.Conexion
Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
Dim cmd As SqlCommand = New
SqlCommand("proc_ModificarMatriculas", cn)
Dim Parametros As New SqlParameter()
'Identificador de la matrícula
Parametros = cmd.Parameters.Add("@IdMatricula", SqlDbType.Char, 8)
Parametros.Value = FilaActual.SubItems(0).Text
'Observación

Parametros = cmd.Parameters.Add("@Observacion", SqlDbType.Char, 1000)
Parametros.Value = Trim(FilaActual.SubItems(12).Text) & "
ANULADA/RETIRADA EL " & FechaAux
cmd.CommandType = CommandType.StoredProcedure
cn.Open()
Dim da As New Data.SqlClient.SqlDataAdapter(cmd)
Dim ds As New DataSet()
da.Fill(ds)
ds.Dispose()
da.Dispose()
cmd.Dispose()
cn.Close()
cn.Dispose()
MsgBox("Matrícula Anulada/Retirada", MsgBoxStyle.Exclamation)
Catch ex As Exception
System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub

Private Sub lvMatriculas_DoubleClick(ByVal sender As Object, ByVal e As
System.EventArgs) Handles lvMatriculas.DoubleClick
Dim FilaActual As ListViewItem
Try
If Me.optModificar.Checked = True Then
Dim frmModificarMatriculas As New frmModificarMatriculas
With frmModificarMatriculas
FilaActual = Me.lvMatriculas.SelectedItems(0)
.txtIdMatricula.Text = FilaActual.SubItems(0).Text
.txtIdEstudiante.Text = Trim(FilaActual.SubItems(1).Text)
.txtIdAula.Text = Trim(FilaActual.SubItems(2).Text)
.cboTurno.Text = Trim(FilaActual.SubItems(3).Text)
.txtAnyoLectivo.Text = Trim(FilaActual.SubItems(4).Text)
```



```
.cboGradoMatricula.Text = Trim(FilaActual.SubItems(5).Text)
.dtpFechaMatricula.Value = Trim(FilaActual.SubItems(6).Text)
.cboPartidaNac.Text = Trim(FilaActual.SubItems(7).Text)
.cboNotasUltimoGrado.Text = Trim(FilaActual.SubItems(8).Text)
.cboDipSextoGrado.Text = Trim(FilaActual.SubItems(9).Text)
.txtOtrosDoc.Text = Trim(FilaActual.SubItems(10).Text)
.cboRepitente.Text = Trim(FilaActual.SubItems(11).Text)
.txtObservacion.Text = Trim(FilaActual.SubItems(12).Text)
.ShowDialog()
```

End With

End If

```
If Me.optAnularRetirar.Checked = True Then
```

```
    Call Me.AnularRetirarMatricula()
```

End If

```
Catch ex As Exception
```

```
    System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
```

End Try

End Sub

End Class

frmModificarMatriculas

```
Public Class frmModificarMatriculas
```

```
    Sub CrearListViewAula()
```

```
        With Me.lvAulas
```

```
            .Clear()
```

```
            .LabelEdit = True
```

```
            .AllowColumnReorder = True
```

```
            .FullRowSelect = True
```

```
            .Sorting = SortOrder.Ascending
```

```
            .View = View.Details
```

```
            .GridLines = True
```

```
            .Columns.Add("Id", 40, HorizontalAlignment.Center)
```

```
            .Columns.Add("Nombre", 200, HorizontalAlignment.Center)
```

```
            .Columns.Add("Descripción", 250, HorizontalAlignment.Center)
```

End With

End Sub

```
Sub CargarAulas()
```

```
    Dim CadenaConexion As String
```

```
    CadenaConexion = modUtilidades.Conexion
```

```
    Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
```

```
    Dim vListarCargos As New Data.SqlClient.SqlCommand("proc_ListarAulas", cn)
```

```
    vListarCargos.CommandType = CommandType.StoredProcedure
```



```
Dim drListarCargos As Data.SqlClient.SqlDataReader
cn.Open()
drListarCargos = vListarCargos.ExecuteReader
'Llenar el ListView
Dim vpos As Integer
Me.lvAulas.Items.Clear()
While drListarCargos.Read
    With Me.lvAulas
        .Items.Add(drListarCargos("idAula")) ' Identificador
        .Items(vpos).SubItems.Add(drListarCargos("Nombre")) ' Nombre
        .Items(vpos).SubItems.Add(drListarCargos("Descripcion")) ' Descripcion
        ' Llenar el combo con los nombres de las aulas
        Me.cboNombreAulas.Items.Add(Trim(drListarCargos("Nombre"))) ' Nombre
        vpos += 1
    End With
End While
If Me.cboNombreAulas.Items.Count = 0 Then
    Me.cmdPasarIdAula.Enabled = False
    Me.cboNombreAulas.Enabled = False
    Me.cmdPasarIdAula.Focus()
Else
    Me.cmdPasarIdAula.Enabled = True
    Me.cmdCancelarPasarIdAula.Enabled = True
    Me.cboNombreAulas.Enabled = True
    Me.cboNombreAulas.SelectedIndex = 0
    Me.cmdPasarIdAula.Focus()
End If
'Cerrando objetos
drListarCargos.Close()
cn.Close()
End Sub

Sub AulasXNombre()
    Dim CadenaConexion As String
    Dim f As Integer
    Try
        CadenaConexion = modUtilidades.Conexion
        Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
        Dim ocmd As SqlCommand = New
SqlClient.SqlCommand("proc_AulasXNombre", cn)
        Dim oParametros As New SqlParameter()
        'nombre del aula a buscar
        If Me.cboNombreAulas.Text = "" Then
            Call Me.CargarAulas()
        End If
    End Try
End Sub
```



Else

```
oParametros = ocmd.Parameters.Add("@nombre", SqlDbType.Char, 250)
oParametros.Value = Me.cboNombreAulas.Text
ocmd.CommandType = CommandType.StoredProcedure
cn.Open()
Dim oda As New Data.SqlClient.SqlDataAdapter(ocmd)
Dim odt As New DataTable()
oda.Fill(odt)
' Creamos un objeto DataRow y nos ubicamos en el 1er registro devuelto
Dim DatosTabla As Data.DataRow
```

```
If odt.Rows.Count = 0 Then
```

```
Exit Sub
```

```
Else
```

```
f = odt.Rows.Count - 1
```

```
DatosTabla = odt.Rows(f)
```

```
Call Me.CrearListViewAula()
```

```
f = 0
```

```
While f <= odt.Rows.Count - 1
```

```
    DatosTabla = odt.Rows(f) ' nos movemos por los regs resultantes
```

```
    Me.lvAulas.Items.Add(DatosTabla("idAula"))
```

```
    Me.lvAulas.Items(f).SubItems.Add(DatosTabla("Nombre"))
```

```
    Me.lvAulas.Items(f).SubItems.Add(DatosTabla("Descripcion"))
```

```
    f += 1
```

```
End While
```

```
odt.Dispose()
```

```
oda.Dispose()
```

```
ocmd.Dispose()
```

```
cn.Close()
```

```
cn.Dispose()
```

```
End If
```

```
End If
```

```
Catch ex As Exception
```

```
    System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
```

```
End Try
```

```
End Sub
```

```
Sub CrearListView()
```

```
    'Formatear el control ListView
```

```
    With Me.lvEstudiantes
```

```
        .Clear()
```

```
        .LabelEdit = True
```

```
        .AllowColumnReorder = True
```



```
.FullRowSelect = True
.Sorting = SortOrder.Ascending
.View = View.Details
.GridLines = True
.Columns.Add("Id", 40, HorizontalAlignment.Center)
.Columns.Add("Nombres", 150, HorizontalAlignment.Center)
.Columns.Add("Apellidos", 150, HorizontalAlignment.Center)
.Columns.Add("Lugar Nac.", 100, HorizontalAlignment.Center)
.Columns.Add("Fecha Nac.", 100, HorizontalAlignment.Center)
.Columns.Add("Edad", 50, HorizontalAlignment.Center)
.Columns.Add("Sexo", 50, HorizontalAlignment.Center)
.Columns.Add("Tipo Sangre", 80, HorizontalAlignment.Center)
.Columns.Add("Dirección", 200, HorizontalAlignment.Center)
.Columns.Add("Teléfono", 100, HorizontalAlignment.Center)
.Columns.Add("Foto", 100, HorizontalAlignment.Center)
```

End With

End Sub

Sub CargarEstudiantes()

Dim CadenaConexion As String

CadenaConexion = modUtilidades.Conexion

Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)

Dim vListarCargos As New Data.SqlClient.SqlCommand("proc_ListarEstudiantes",

cn)

vListarCargos.CommandType = CommandType.StoredProcedure

Dim drListarCargos As Data.SqlClient.SqlDataReader

cn.Open()

drListarCargos = vListarCargos.ExecuteReader

'Llenar el ListView

Dim vpos As Integer

Me.lvEstudiantes.Items.Clear()

While drListarCargos.Read

With Me.lvEstudiantes

.Items.Add(drListarCargos("idEstudiante")) ' Identificador

.Items(vpos).SubItems.Add(drListarCargos("Nombres")) ' Nombres

.Items(vpos).SubItems.Add(drListarCargos("Apellidos")) ' Apellidos

.Items(vpos).SubItems.Add(drListarCargos("LugarNacimiento")) ' Lugar de

nacimiento

.Items(vpos).SubItems.Add(drListarCargos("FechaNacimiento")) ' Fecha de

nacimiento

.Items(vpos).SubItems.Add(drListarCargos("Edad")) ' Edad

.Items(vpos).SubItems.Add(drListarCargos("Sexo")) ' Sexo

.Items(vpos).SubItems.Add(drListarCargos("TipoSangre")) ' Tipo de sangre

.Items(vpos).SubItems.Add(drListarCargos("Direccion")) ' Dirección



```
.Items(vpos).SubItems.Add(drListarCargos("Telefono")) ' Teléfono
.Items(vpos).SubItems.Add(drListarCargos("Foto")) ' Foto
' Llenar el combo con los nombres de las aulas
Me.cboApellidosEstudiantes.Items.Add(Trim(drListarCargos("Apellidos"))) '
Apellidos
    vpos += 1
End With
End While
If Me.cboApellidosEstudiantes.Items.Count = 0 Then
    Me.cmdAceptar.Enabled = False
    Me.cboApellidosEstudiantes.Enabled = False
    Me.cmdCancelar.Focus()
Else
    Me.cmdAceptar.Enabled = True
    Me.cmdCancelar.Enabled = True
    Me.cboApellidosEstudiantes.Enabled = True
    Me.cboApellidosEstudiantes.SelectedIndex = 0
    Me.cmdAceptar.Focus()
End If
'Cerrando objetos
drListarCargos.Close()
cn.Close()
End Sub

Sub EstudiantesXApellidos()
    Dim CadenaConexion As String
    Dim f As Integer
    Try
        CadenaConexion = modUtilidades.Conexion
        Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
        Dim ocmd As SqlCommand = New
SqlClient.SqlCommand("proc_EstudiantesXApellidos", cn)
        Dim oParametros As New SqlParameter()
        'Apellidos del estudiante a buscar
        If Me.cboApellidosEstudiantes.Text = "" Then
            Call Me.CargarEstudiantes()
        Else
            oParametros = ocmd.Parameters.Add("@apellidos", SqlDbType.Char, 250)
            oParametros.Value = Me.cboApellidosEstudiantes.Text
            ocmd.CommandType = CommandType.StoredProcedure
            cn.Open()
            Dim oda As New SqlDataAdapter(ocmd)
            Dim odt As New DataTable()
            oda.Fill(odt)
```



```
' Creo un objeto DataRow y nos ubicamos en el 1er registro devuelto
Dim DatosTabla As Data.DataRow
If odt.Rows.Count = 0 Then
    Exit Sub
Else
    f = odt.Rows.Count - 1
    DatosTabla = odt.Rows(f)
    Call Me.CrearListView()
    f = 0
    While f <= odt.Rows.Count - 1
        DatosTabla = odt.Rows(f) ' me muevo por los regs resultantes
        Me.lvEstudiantes.Items.Add(DatosTabla("idEstudiante"))

        Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("Nombres"))
        Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("Apellidos"))
        Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("LugarNacimiento"))
        Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("FechaNacimiento"))
        Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("Edad"))
        Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("Sexo"))
        Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("TipoSangre"))
        Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("Direccion"))
        Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("Telefono"))
        Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("Foto"))
        f += 1
    End While
    odt.Dispose()
    oda.Dispose()
    ocmd.Dispose()
    cn.Close()
    cn.Dispose()
End If
End If
Catch ex As Exception
    System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub

Sub LlenarCombos()
    With Me.cboTurno
        .Items.Add("Matutino")
        .Items.Add("Vespertino")
    End With
    With Me.cboGradoMatricula
        .Items.Add("I")
        .Items.Add("II")
    End With
End Sub
```



```
.Items.Add("III")  
.Items.Add("IV")  
.Items.Add("V")
```

End With

With Me.cboRepitente

```
.Items.Add("Si")  
.Items.Add("No")
```

End With

With Me.cboPartidaNac

```
.Items.Add("Si")  
  
.Items.Add("No")
```

End With

With Me.cboNotasUltimoGrado

```
.Items.Add("Si")  
.Items.Add("No")
```

End With

With Me.cboDipSextoGrado

```
.Items.Add("Si")  
.Items.Add("No")
```

End With

End Sub

Sub ModificarMatriculas()

Dim CadenaConexion As String

Try

CadenaConexion = modUtilidades.Conexion

Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)

Dim cmd As SqlCommand = New

SqlCommand("proc_ModificarDatosMatricula", cn)

Dim Parametros As New SqlParameter()

'Identificador de la matricula

Parametros = cmd.Parameters.Add("@IdMatricula", SqlDbType.Char, 8)

Parametros.Value = Me.txtIdMatricula.Text

'Identificador del estudiante

Parametros = cmd.Parameters.Add("@IdEstudiante", SqlDbType.Char, 8)

Parametros.Value = Me.txtIdEstudiante.Text

'Identificador del aula

Parametros = cmd.Parameters.Add("@IdAula", SqlDbType.Char, 4)

Parametros.Value = Me.txtIdAula.Text



'Turno

Parametros = cmd.Parameters.Add("@Turno", SqlDbType.Char, 10)

Parametros.Value = Trim(Me.cboTurno.Text)

'Año lectivo

Parametros = cmd.Parameters.Add("@AnyoLectivo", SqlDbType.Int)

Parametros.Value = Me.txtAnyoLectivo.Text

'Grado matrícula

Parametros = cmd.Parameters.Add("@GradoMatricula", SqlDbType.Char, 5)

Parametros.Value = Trim(Me.cboGradoMatricula.Text)

'Fecha matrícula

Parametros = cmd.Parameters.Add("@FechaMatricula", SqlDbType.DateTime)

Parametros.Value = Me.dtpFechaMatricula.Value

'Partida de nacimiento

Parametros = cmd.Parameters.Add("@PartidaNacimiento", SqlDbType.Char, 2)

Parametros.Value = Trim(Me.cboPartidaNac.Text)

'Notas del último grado

Parametros = cmd.Parameters.Add("@NotasUltimoGrado", SqlDbType.Char, 2)

Parametros.Value = Trim(Me.cboNotasUltimoGrado.Text)

'Diploma de sexto grado

Parametros = cmd.Parameters.Add("@DiplomaSextoGrado", SqlDbType.Char,

2)

Parametros.Value = Trim(Me.cboDipSextoGrado.Text)

'Otros documentos

Parametros = cmd.Parameters.Add("@OtrosDocumentos", SqlDbType.Char,

500)

Parametros.Value = Trim(Me.txtOtrosDoc.Text)

'Repitente

Parametros = cmd.Parameters.Add("@Repitente", SqlDbType.Char, 2)

Parametros.Value = Trim(Me.cboRepitente.Text)

'Observación

Parametros = cmd.Parameters.Add("@Observacion", SqlDbType.Char, 1000)

Parametros.Value = Trim(Me.txtObservacion.Text)

cmd.CommandType = CommandType.StoredProcedure

cn.Open()

Dim da As New Data.SqlClient.SqlDataAdapter(cmd)

Dim ds As New Data.DataSet()

da.Fill(ds)

ds.Dispose()

da.Dispose()

cmd.Dispose()

cn.Close()

cn.Dispose()

MsgBox("Matrícula modificada", MsgBoxStyle.Exclamation)



```
Catch ex As Exception
    System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub
```

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

```
    Me.Hide()
    Me.Close()
End Sub
```

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

```
    Me.grpEstudiantes.Visible = False
End Sub
```

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

```
    Dim FilaActual As ListViewItem
    Try
        FilaActual = Me.lvEstudiantes.SelectedItems(0)
        Me.txtIdEstudiante.Text = FilaActual.SubItems(0).Text
        Me.grpEstudiantes.Visible = False
    Catch ex As Exception
        System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
    End Try
End Sub
```

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

```
    Dim FilaActual As ListViewItem
    Try
        FilaActual = Me.lvAulas.SelectedItems(0)
        Me.txtIdAula.Text = FilaActual.SubItems(0).Text
        Me.grpAulas.Visible = False
    Catch ex As Exception
        System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
    End Try
End Sub
```

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



```
Me.grpAulas.Visible = False  
End Sub
```

```
Private Sub frmModificarMatriculas_Load(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles MyBase.Load  
    Call Me.LlenarCombos()  
End Sub
```

```
Private Sub cmdBuscarEstudiantes_Click(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles cmdBuscarEstudiantes.Click  
    Me.grpEstudiantes.Visible = True  
    Me.grpAulas.Visible = False  
  
    Call Me.CrearListView()  
    Call Me.CargarEstudiantes()  
    Me.cboApellidosEstudiantes.Focus()  
End Sub
```

```
Private Sub cmdBuscarAulas_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles cmdBuscarAulas.Click  
    Me.grpAulas.Visible = True  
    Me.grpEstudiantes.Visible = False  
    Call Me.CrearListViewAula()  
    Call Me.CargarAulas()  
    Me.cboNombreAulas.Focus()  
End Sub
```

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

```
    If Len(Me.txtIdEstudiante.Text) = 0 Then  
        MsgBox("Debe especificar el identificador del estudiante", MsgBoxStyle.Critical,  
"INSS")  
        Me.txtIdEstudiante.Focus()  
        Exit Sub  
    ElseIf Len(Me.txtIdAula.Text) = 0 Then  
        MsgBox("Debe especificar el identificador del aula", MsgBoxStyle.Critical,  
"INSS")  
        Me.txtIdAula.Focus()  
        Exit Sub  
    ElseIf Len(Me.cboTurno.Text) = 0 Then  
        MsgBox("Debe especificar el turno de la matrícula", MsgBoxStyle.Critical,  
"INSS")
```



```
Me.cboTurno.Focus()
Exit Sub
Elseif Len(Me.txtAnyoLectivo.Text) = 0 Then
    MsgBox("Debe especificar el año lectivo", MsgBoxStyle.Critical, "INSS")
    Me.txtAnyoLectivo.Focus()
    Exit Sub
Elseif Len(Me.cboGradoMatricula.Text) = 0 Then
    MsgBox("Debe especificar el grado de la matrícula", MsgBoxStyle.Critical,
"INSS")
    Me.cboGradoMatricula.Focus()
    Exit Sub
Elseif Len(Me.cboRepitente.Text) = 0 Then
    MsgBox("Debe especificar si es repitente o no", MsgBoxStyle.Critical, "INSS")

    Me.cboRepitente.Focus()
    Exit Sub
Elseif Len(Me.cboPartidaNac.Text) = 0 Then
    MsgBox("Debe especificar la partida de nacimiento", MsgBoxStyle.Critical,
"INSS")
    Me.cboPartidaNac.Focus()
    Exit Sub
Elseif Len(Me.cboNotasUltimoGrado.Text) = 0 Then
    MsgBox("Debe especificar notas del último año", MsgBoxStyle.Critical, "INSS")
    Me.cboNotasUltimoGrado.Focus()
    Exit Sub
Elseif Len(Me.cboDipSextoGrado.Text) = 0 Then
    MsgBox("Debe especificar si trajo el diploma de sexto grado",
MsgBoxStyle.Critical, "INSS")
    Me.cboDipSextoGrado.Focus()
    Exit Sub
Else
    Call Me.ModificarMatriculas()
    Me.Hide()
    Me.Close()
End If
End Sub

Private Sub txtAnyoLectivo_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtAnyoLectivo.KeyPress

    If (Asc(e.KeyChar)) >= 48 And (Asc(e.KeyChar)) <= 57 Or _
(Asc(e.KeyChar)) = System.Windows.Forms.Keys.Back Then
        e.Handled = False
```



```
Else  
    e.Handled = True  
End If  
End Sub
```

```
Private Sub txtOtrosDoc_KeyPress(ByVal sender As Object, ByVal e As  
System.Windows.Forms.KeyPressEventArgs) Handles txtOtrosDoc.KeyPress
```

```
    If (Asc(e.KeyChar)) >= 65 And (Asc(e.KeyChar)) <= 90 Or _  
        (Asc(e.KeyChar)) >= 97 And (Asc(e.KeyChar)) <= 122 Or _  
        (Asc(e.KeyChar)) = System.Windows.Forms.Keys.Back Or _  
        (Asc(e.KeyChar)) = 44 Or (Asc(e.KeyChar)) = 46 Or _  
        (Asc(e.KeyChar)) = 241 Or (Asc(e.KeyChar)) = 209 Or _  
        (Asc(e.KeyChar)) = 160 Or (Asc(e.KeyChar)) = 130 Or _  
  
        (Asc(e.KeyChar)) = 161 Or (Asc(e.KeyChar)) = 162 Or _  
        (Asc(e.KeyChar)) = 163 Or (Asc(e.KeyChar)) = 129 Or _  
        (Asc(e.KeyChar)) = 154 Or _  
        (Asc(e.KeyChar)) = 32 Then  
        e.Handled = False  
    Else  
        e.Handled = True  
    End If  
End Sub
```

```
Private Sub txtObservacion_KeyPress(ByVal sender As Object, ByVal e As  
System.Windows.Forms.KeyPressEventArgs) Handles txtObservacion.KeyPress
```

```
    If (Asc(e.KeyChar)) >= 65 And (Asc(e.KeyChar)) <= 90 Or _  
        (Asc(e.KeyChar)) >= 97 And (Asc(e.KeyChar)) <= 122 Or _  
        (Asc(e.KeyChar)) = System.Windows.Forms.Keys.Back Or _  
        (Asc(e.KeyChar)) = 44 Or (Asc(e.KeyChar)) = 46 Or _  
        (Asc(e.KeyChar)) = 241 Or (Asc(e.KeyChar)) = 209 Or _  
        (Asc(e.KeyChar)) = 160 Or (Asc(e.KeyChar)) = 130 Or _  
        (Asc(e.KeyChar)) = 161 Or (Asc(e.KeyChar)) = 162 Or _  
        (Asc(e.KeyChar)) = 163 Or (Asc(e.KeyChar)) = 129 Or _  
        (Asc(e.KeyChar)) = 154 Or _  
        (Asc(e.KeyChar)) = 32 Then  
        e.Handled = False  
    Else  
        e.Handled = True  
    End If  
End Sub
```



End Class

frmCatEstudiantes

Public Class frmCatEstudiantes

Sub CrearListView()

'Formatear el control ListView

With Me.lvEstudiantes

.Clear()

.LabelEdit = False

.AllowColumnReorder = True

.FullRowSelect = True

.Sorting = SortOrder.Ascending

.View = View.Details

.GridLines = True

.Columns.Add("Id. Est.", 80, HorizontalAlignment.Center)

.Columns.Add("Nombres", 150, HorizontalAlignment.Center)

.Columns.Add("Apellidos", 150, HorizontalAlignment.Center)

.Columns.Add("Lugar Nac.", 100, HorizontalAlignment.Center)

.Columns.Add("Fecha Nac.", 100, HorizontalAlignment.Center)

.Columns.Add("Edad", 50, HorizontalAlignment.Center)

.Columns.Add("Sexo", 50, HorizontalAlignment.Center)

.Columns.Add("Tipo Sangre", 80, HorizontalAlignment.Center)

.Columns.Add("Dirección", 200, HorizontalAlignment.Center)

.Columns.Add("Teléfono", 100, HorizontalAlignment.Center)

End With

End Sub

Sub CargarEstudiantes()

Dim CadenaConexion As String

CadenaConexion = modUtilidades.Conexion

Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)

Dim vListarCargos As New Data.SqlClient.SqlCommand("proc_ListarEstudiantes",

cn)

vListarCargos.CommandType = CommandType.StoredProcedure

Dim drListarCargos As Data.SqlClient.SqlDataReader

cn.Open()

drListarCargos = vListarCargos.ExecuteReader

'Llenar el ListView

Dim vpos As Integer

Me.lvEstudiantes.Items.Clear()

While drListarCargos.Read



```
With Me.lvEstudiantes
.Items.Add(drListarCargos("idEstudiante")) ' Identificador
.Items(vpos).SubItems.Add(drListarCargos("Nombres")) ' Nombres
.Items(vpos).SubItems.Add(drListarCargos("Apellidos")) ' Apellidos
.Items(vpos).SubItems.Add(drListarCargos("LugarNacimiento")) ' Lugar de
nacimiento
.Items(vpos).SubItems.Add(drListarCargos("FechaNacimiento")) ' Fecha de
nacimiento
.Items(vpos).SubItems.Add(drListarCargos("Edad")) ' Edad
.Items(vpos).SubItems.Add(drListarCargos("Sexo")) ' Sexo
.Items(vpos).SubItems.Add(drListarCargos("TipoSangre")) ' Tipo de sangre
.Items(vpos).SubItems.Add(drListarCargos("Direccion")) ' Dirección
.Items(vpos).SubItems.Add(drListarCargos("Telefono")) ' Teléfono
' Llenar el combo con los nombres de las aulas
Me.cboApellidosEstudiantes.Items.Add(Trim(drListarCargos("Apellidos"))) '
Apellidos
vpos += 1
End With
End While
If Me.cboApellidosEstudiantes.Items.Count = 0 Then
Me.cmdBorrar.Enabled = False
Me.cmdModificar.Enabled = False
Me.cboApellidosEstudiantes.Enabled = False
Me.cmdNuevo.Focus()
Else
Me.cmdBorrar.Enabled = True
Me.cmdModificar.Enabled = True
Me.cboApellidosEstudiantes.Enabled = True
Me.cboApellidosEstudiantes.SelectedIndex = 0
Me.cmdNuevo.Focus()
End If

'Cerrando objetos
drListarCargos.Close()
cn.Close()
End Sub

Sub EstudiantesXApellidos()
Dim CadenaConexion As String
Dim f As Integer
Try
CadenaConexion = modUtilidades.Conexion
Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
```



```
Dim ocmd As SqlConnection.SqlCommand = New
SqlConnection.SqlCommand("proc_EstudiantesXApellidos", cn)
Dim oParametros As New SqlConnection.SqlParameter()
'Apellidos del estudiante a buscar
If Me.cboApellidosEstudiantes.Text = "" Then
    Exit Sub
Else
    oParametros = ocmd.Parameters.Add("@apellidos", SqlDbType.Char, 250)
    oParametros.Value = Me.cboApellidosEstudiantes.Text
    ocmd.CommandType = CommandType.StoredProcedure
    cn.Open()
    Dim oda As New Data.SqlClient.SqlDataAdapter(ocmd)
    Dim odt As New DataTable()
    oda.Fill(odt)
    ' Creo un objeto DataRow y nos ubicamos en el 1er registro devuelto
    Dim DatosTabla As DataRow
    If odt.Rows.Count = 0 Then
        Call Me.CrearListView()
        Exit Sub
    Else
        f = odt.Rows.Count - 1
        DatosTabla = odt.Rows(f)
        Call Me.CrearListView()
        f = 0
        While f <= odt.Rows.Count - 1
            DatosTabla = odt.Rows(f) ' me muevo por los regs resultantes
            Me.lvEstudiantes.Items.Add(DatosTabla("idEstudiante"))
            Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("Nombres"))
            Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("Apellidos"))
            Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("LugarNacimiento"))
            Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("FechaNacimiento"))
            Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("Edad"))
            Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("Sexo"))
            Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("TipoSangre"))
            Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("Direccion"))
            Me.lvEstudiantes.Items(f).SubItems.Add(DatosTabla("Telefono"))
            f += 1
        End While
        odt.Dispose()
        oda.Dispose()
        ocmd.Dispose()
        cn.Close()
        cn.Dispose()
    End If
End If
```



```
End If
Catch ex As Exception
    System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub
```

```
Private Sub frmCatEstudiantes_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Call Me.CrearListView()
    Call Me.CargarEstudiantes()
End Sub
```

```
Private Sub cmdCancelar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdCancelar.Click
    Me.Close()
End Sub
```

```
Private Sub cboApellidosEstudiantes_KeyUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles cboApellidosEstudiantes.KeyUp
    Call Me.EstudiantesXApellidos()
End Sub
```

```
Private Sub cmdModificar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdModificar.Click
    Dim FilaActual As ListViewItem
    Dim IdAux As String
    Dim NombresAux As String
    Dim ApellidosAux As String
    Dim LugarNacimientoAux As String
    Dim FechaNacimientoAux As String
    Dim EdadAux As String
    Dim SexoAux As String
    Dim TipoSangreAux As String
    Dim DireccionAux As String
    Dim TelefonoAux As String
    Try
        If Me.lvEstudiantes.SelectedItems.Count = 0 Then
            MsgBox("Debe especificar el estudiante que desea modificar",
MsgBoxStyle.Exclamation)
            Exit Sub
        Else
            FilaActual = Me.lvEstudiantes.SelectedItems(0)
```



```
IdAux = FilaActual.SubItems(0).Text
NombresAux = Trim(FilaActual.SubItems(1).Text)
ApellidosAux = Trim(FilaActual.SubItems(2).Text)
LugarNacimientoAux = FilaActual.SubItems(3).Text
FechaNacimientoAux = FilaActual.SubItems(4).Text
EdadAux = FilaActual.SubItems(5).Text
SexoAux = FilaActual.SubItems(6).Text
TipoSangreAux = FilaActual.SubItems(7).Text
DireccionAux = FilaActual.SubItems(8).Text
TelefonoAux = FilaActual.SubItems(9).Text
With frmModificarEstudiantes
    .txtIdentificador.Text = IdAux
    .txtNombres.Text = NombresAux
    .txtApellidos.Text = ApellidosAux
    .cboLugarNacimiento.Text = LugarNacimientoAux
    .dtpFechaNacimiento.Value = FechaNacimientoAux
    .txtEdad.Text = EdadAux
    .cboSexo.Text = SexoAux
    .cboTipoSangre.Text = TipoSangreAux
    .txtDireccion.Text = DireccionAux

    .mskTelefono.Text = TelefonoAux
    Me.Hide()
    Me.Close()
    .Show()
End With
End If

Catch ex As Exception
    System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub

Sub EliminarEstudiante()
    Dim IdAux As String
    Dim CadenaConexion As String
    Dim FilaActual As ListViewItem
    Try
        'Obtener el Identificador del aula a borrar
        FilaActual = Me.lvEstudiantes.SelectedItem(0)
        IdAux = FilaActual.SubItems(0).Text
        CadenaConexion = modUtilidades.Conexion
        Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
```



```
Dim ocmd As SqlConnection.SqlCommand = New
SqlConnection.SqlCommand("proc_EliminarEstudiante", cn)

Dim oParametros As New SqlConnection.SqlParameter()
'Identificador
oParametros = ocmd.Parameters.Add("@IdEstudiante", SqlDbType.Char, 8)
oParametros.Value = IdAux
ocmd.CommandType = CommandType.StoredProcedure
cn.Open()
Dim oda As New Data.SqlClient.SqlDataAdapter(ocmd)
Dim ods As New Data.DataSet()
oda.Fill(ods)
'Cerramos o liberamos recursos
ods.Dispose()
oda.Dispose()
ocmd.Dispose()
cn.Close()
cn.Dispose()
MsgBox("Estudiante eliminado", MsgBoxStyle.Exclamation, "Sistema INSS")
Me.Hide()

Me.Close()
Catch ex As Exception
System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try

End Sub

Private Sub cmdBorrar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdBorrar.Click
Try
If Me.lvEstudiantes.SelectedItems.Count = 0 Then

MsgBox("Debe especificar el estudiante a borrar", MsgBoxStyle.Exclamation)
Exit Sub
Else
Call Me.EliminarEstudiante()
Dim frmCatEstudiantes As New frmCatEstudiantes
frmCatEstudiantes.ShowDialog()
End If
Catch ex As Exception
System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub
```



```
Private Sub cmdNuevo_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdNuevo.Click
    Me.Hide()

    Me.Close()
    frmNuevoEstudiante.ShowDialog()
End Sub
End Class
```

frmNuevoEstudiante

```
Public Class frmNuevoEstudiante
    Public Edad As Integer
    Sub UltimoCodigoEstudiante()
        Dim CadenaConexion As String
        CadenaConexion = modUtilidades.Conexion
        Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
        Dim vUltimoCodigo As New Data.SqlClient.SqlCommand("proc_GenerarIdEstudiante", cn)
        vUltimoCodigo.CommandType = CommandType.StoredProcedure
        Dim drUltimoCodigo As Data.SqlClient.SqlDataReader
        cn.Open()
        drUltimoCodigo = vUltimoCodigo.ExecuteReader
        If drUltimoCodigo.Read Then
            Me.txtIdentificador.Text = Format(drUltimoCodigo("MaxCodigo"), "00000000")
        Else
            Me.txtIdentificador.Text = Format("1", "00000000")
        End If
        'Cerrando objetos
        drUltimoCodigo.Close()
        cn.Close()
    End Sub

    Sub ValidarEntradas()
        If Me.txtNombres.Text = "" Then
            MsgBox("Debe especificar los nombres del estudiante",
            MsgBoxStyle.Exclamation)
            Me.txtNombres.Focus()
            Exit Sub
        End If
        If Me.txtApellidos.Text = "" Then
            MsgBox("Debe especificar los apellidos del estudiante",
            MsgBoxStyle.Exclamation)
            Me.txtApellidos.Focus()
        End If
    End Sub
End Class
```



```
Exit Sub
End If
If Me.cboLugarNacimiento.Text = "" Then
    MsgBox("Debe especificar el lugar de nacimiento del estudiante",
MsgBoxStyle.Exclamation)
    Me.cboLugarNacimiento.Focus()
Exit Sub
End If
If Me.dtpFechaNacimiento.Text = "" Then
    MsgBox("Debe especificar la fecha de nacimiento del estudiante",
MsgBoxStyle.Exclamation)
    Me.dtpFechaNacimiento.Focus()
Exit Sub
End If
If Me.txtEdad.Text = "" Then
    MsgBox("Debe especificar la edad del estudiante", MsgBoxStyle.Exclamation)
    Me.txtEdad.Focus()
Exit Sub
End If
If Me.cboSexo.Text = "" Then
    MsgBox("Debe especificar el sexo del estudiante", MsgBoxStyle.Exclamation)
    Me.cboSexo.Focus()
Exit Sub
End If
If Me.cboTipoSangre.Text = "" Then
    MsgBox("Debe especificar el tipo de sangre del estudiante",
MsgBoxStyle.Exclamation)
    Me.cboTipoSangre.Focus()
Exit Sub
End If
If Me.txtDireccion.Text = "" Then
    MsgBox("Debe especificar al dirección del estudiante",
MsgBoxStyle.Exclamation)
    Me.txtDireccion.Focus()
Exit Sub
End If
If Me.mskTelefono.Text = "" Then
    MsgBox("Debe especificar el teléfono del estudiante",
MsgBoxStyle.Exclamation)
    Me.mskTelefono.Focus()
Exit Sub
End If
End Sub
```



```
Sub LlenarCombosYMascara()  
  With Me.cboLugarNacimiento  
    .Text = ""  
    .Items.Add("Managua")  
    .Items.Add("León")  
    .Items.Add("Masaya")  
    .Items.Add("Granada")  
    .Items.Add("Corinto")  
    .Items.Add("Chinandega")  
    .Items.Add("El Viejo")  
    .Items.Add("Rivas")  
    .Items.Add("Telica")  
    .Items.Add("Quezalguaque")  
    .Items.Add("Boaco")  
    .Items.Add("Carazo")  
    .Items.Add("Chontales")  
    .Items.Add("Estelí")  
    .Items.Add("Jinotega")  
    .Items.Add("Matagalpa")  
    .Items.Add("Madriz")  
    .Items.Add("Nueva Segobia")  
    .Items.Add("Río San Juan")  
  
    .Items.Add("Atlántico Norte")  
    .Items.Add("Atlántico Sur")  
    .Items.Add("Otro")  
    .SelectedIndex = 0  
  End With  
  With Me.cboSexo  
    .Text = ""  
    .Items.Add("M")  
    .Items.Add("F")  
    .SelectedIndex = 0  
  End With  
  With Me.cboTipoSangre  
    .Items.Add("A+")  
    .Items.Add("A-")  
    .Items.Add("O+")  
    .Items.Add("O-")  
    .Items.Add("AB+")  
    .Items.Add("AB-")  
    .SelectedIndex = 0  
  End With
```



```
With Me.mskTelefono
```

```
.Text = ""
```

```
.Mask = ""
```

```
.Mask = "###-####"
```

```
End With
```

```
End Sub
```

```
Sub InsertarNuevoEstudiante()
```

```
Dim CadenaConexion As String
```

```
Try
```

```
    CadenaConexion = modUtilidades.Conexion
```

```
    Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
```

```
    Dim cmd As SqlCommand = New
```

```
    SqlCommand("proc_InsertarEstudiante", cn)
```

```
    Dim Parametros As New SqlParameter()
```

```
        'identificador
```

```
        Parametros = cmd.Parameters.Add("@IdEstudiante", SqlDbType.Char, 8)
```

```
        Parametros.Value = Me.txtIdentificador.Text
```

```
        'nombres
```

```
        Parametros = cmd.Parameters.Add("@Nombres", SqlDbType.Char, 250)
```

```
        Parametros.Value = Me.txtNombres.Text
```

```
        'apellidos
```

```
        Parametros = cmd.Parameters.Add("@Apellidos", SqlDbType.Char, 250)
```

```
        Parametros.Value = Me.txtApellidos.Text
```

```
        'lugar de nacimiento
```

```
        Parametros = cmd.Parameters.Add("@LugarNacimiento", SqlDbType.Char, 250)
```

```
        Parametros.Value = Me.cboLugarNacimiento.Text
```

```
        'fecha de nacimiento
```

```
        Parametros = cmd.Parameters.Add("@FechaNacimiento",  
SqlDbType.DateTime)
```

```
        Parametros.Value = Me.dtpFechaNacimiento.Text
```

```
        'edad
```

```
        Parametros = cmd.Parameters.Add("@Edad", SqlDbType.Int)
```

```
        Parametros.Value = Me.txtEdad.Text
```

```
        'sexo
```

```
        Parametros = cmd.Parameters.Add("@Sexo", SqlDbType.Char, 1)
```

```
        Parametros.Value = Me.cboSexo.Text
```

```
        'tipo de sangre
```

```
        Parametros = cmd.Parameters.Add("@TipoSangre", SqlDbType.Char, 3)
```

```
        Parametros.Value = Me.cboTipoSangre.Text
```

```
        'dirección
```

```
        Parametros = cmd.Parameters.Add("@Direccion", SqlDbType.Char, 500)
```

```
        Parametros.Value = Me.txtDireccion.Text
```



'teléfono

Parametros = cmd.Parameters.Add("@Telefono", SqlDbType.Char, 8)

Parametros.Value = Me.mskTelefono.Text

cmd.CommandType = CommandType.StoredProcedure

cn.Open()

Dim da As New Data.SqlClient.SqlDataAdapter(cmd)

Dim ds As New Data.DataSet()

da.Fill(ds)

'Cerramos o liberamos recursos

ds.Dispose()

da.Dispose()

cmd.Dispose()

cn.Close()

cn.Dispose()

Catch ex As Exception

System.Windows.Forms.MessageBox.Show(ex.Message.ToString)

End Try

End Sub

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

Call Me.UltimoCodigoEstudiante()

Call Me.LlenarCombosYMascara()

End Sub

Private Sub cmdCancelar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdCancelar.Click

Me.Hide()

Me.Close()

frmPrincipal.IsMdiContainer = True 'Se utiliza para especificar que el FrmPadre es un Mdi Padre

frmCatEstudiantes.MdiParent = frmPrincipal 'Indica que es un formulario secundario

frmCatEstudiantes.Show() 'Muestra el formulario

End Sub

Private Sub cmdAceptar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdAceptar.Click

If Len(Me.txtNombres.Text) = 0 Then

MsgBox("Debe especificar los nombres del estudiante", MsgBoxStyle.Critical, "INSS")

Me.txtNombres.Focus()

Exit Sub

Elseif Len(Me.txtApellidos.Text) = 0 Then



```
MsgBox("Debe especificar los apellidos del estudiante", MsgBoxStyle.Critical,
"INSS")
    Me.txtApellidos.Focus()
    Exit Sub
    ElseIf Len(Me.cboLugarNacimiento.Text) = 0 Then
        MsgBox("Debe especificar el lugar de nacimiento del estudiante",
MsgBoxStyle.Critical, "INSS")
        Me.cboLugarNacimiento.Focus()
        Exit Sub
    ElseIf Len(Me.txtEdad.Text) = 0 Then
        MsgBox("Debe especificar la edad del estudiante", MsgBoxStyle.Critical, "INSS")
        Me.txtEdad.Focus()
        Exit Sub
    ElseIf Len(Me.cboSexo.Text) = 0 Then
        MsgBox("Debe especificar el sexo del estudiante", MsgBoxStyle.Critical, "INSS")
        Me.cboSexo.Focus()
        Exit Sub
    ElseIf Len(Me.cboTipoSangre.Text) = 0 Then
        MsgBox("Debe especificar el tipo de sangre del estudiante",
MsgBoxStyle.Critical, "INSS")
        Me.cboTipoSangre.Focus()
        Exit Sub

    ElseIf Len(Me.txtDireccion.Text) = 0 Then

        MsgBox("Debe especificar la dirección del estudiante", MsgBoxStyle.Critical,
"INSS")
        Me.txtDireccion.Focus()
        Exit Sub
    ElseIf Len(Me.mskTelefono.Text) <> 8 Then
        MsgBox("Debe especificar el número de teléfono del estudiante",
MsgBoxStyle.Critical, "INSS")
        Me.mskTelefono.Focus()
        Exit Sub
    Else
        Call Me.UltimoCodigoEstudiante()
        Call Me.InsertarNuevoEstudiante()
        Me.txtNombres.Clear()
        Me.txtApellidos.Clear()
        Me.txtEdad.Clear()
        Me.txtDireccion.Clear()
        Me.mskTelefono.Clear()
        Me.Hide()
        Me.Close()
```



frmPrincipal.IsMdiContainer = True 'Se utiliza para especificar que el FrmPadre es un Mdi Padre

frmCatEstudiantes.MdiParent = frmPrincipal 'Indica que es un formulario secundario

frmCatEstudiantes.Show() 'Muestra el formulario

End If

End Sub

Private Sub txtNombres_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtNombres.KeyPress

```
If (Asc(e.KeyChar)) >= 65 And (Asc(e.KeyChar)) <= 90 Or _
(Asc(e.KeyChar)) >= 97 And (Asc(e.KeyChar)) <= 122 Or _
(Asc(e.KeyChar)) = System.Windows.Forms.Keys.Back Or _
(Asc(e.KeyChar)) = 44 Or (Asc(e.KeyChar)) = 46 Or _
(Asc(e.KeyChar)) = 241 Or (Asc(e.KeyChar)) = 209 Or _
(Asc(e.KeyChar)) = 160 Or (Asc(e.KeyChar)) = 130 Or _
(Asc(e.KeyChar)) = 161 Or (Asc(e.KeyChar)) = 162 Or _
(Asc(e.KeyChar)) = 163 Or (Asc(e.KeyChar)) = 129 Or _
(Asc(e.KeyChar)) = 154 Or _
(Asc(e.KeyChar)) = 32 Then
    e.Handled = False
```

Else

e.Handled = True

End If

End Sub

Private Sub txtApellidos_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtApellidos.KeyPress

```
If (Asc(e.KeyChar)) >= 65 And (Asc(e.KeyChar)) <= 90 Or _
(Asc(e.KeyChar)) >= 97 And (Asc(e.KeyChar)) <= 122 Or _
(Asc(e.KeyChar)) = System.Windows.Forms.Keys.Back Or _
(Asc(e.KeyChar)) = 44 Or (Asc(e.KeyChar)) = 46 Or _
(Asc(e.KeyChar)) = 241 Or (Asc(e.KeyChar)) = 209 Or _
(Asc(e.KeyChar)) = 160 Or (Asc(e.KeyChar)) = 130 Or _
(Asc(e.KeyChar)) = 161 Or (Asc(e.KeyChar)) = 162 Or _
(Asc(e.KeyChar)) = 163 Or (Asc(e.KeyChar)) = 129 Or _
(Asc(e.KeyChar)) = 154 Or _
(Asc(e.KeyChar)) = 32 Then
    e.Handled = False
```

Else

e.Handled = True

End If

End Sub



```
Private Sub txtEdad_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtEdad.KeyPress
    If (Asc(e.KeyChar)) >= 48 And (Asc(e.KeyChar)) <= 57 Or _
        (Asc(e.KeyChar)) = System.Windows.Forms.Keys.Back Then
        e.Handled = False
    Else
        e.Handled = True
    End If
End Sub
```

```
Private Sub txtDireccion_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtDireccion.KeyPress
    If (Asc(e.KeyChar)) >= 65 And (Asc(e.KeyChar)) <= 90 Or _
        (Asc(e.KeyChar)) >= 97 And (Asc(e.KeyChar)) <= 122 Or _
        (Asc(e.KeyChar)) = System.Windows.Forms.Keys.Back Or _
        (Asc(e.KeyChar)) = 44 Or (Asc(e.KeyChar)) = 46 Or _
        (Asc(e.KeyChar)) = 241 Or (Asc(e.KeyChar)) = 209 Or _
        (Asc(e.KeyChar)) = 160 Or (Asc(e.KeyChar)) = 130 Or _
        (Asc(e.KeyChar)) = 161 Or (Asc(e.KeyChar)) = 162 Or _
        (Asc(e.KeyChar)) = 163 Or (Asc(e.KeyChar)) = 129 Or _
        (Asc(e.KeyChar)) >= 48 And (Asc(e.KeyChar)) <= 57 Or _
        (Asc(e.KeyChar)) = 154 Or _
        (Asc(e.KeyChar)) = 47 Or _
        (Asc(e.KeyChar)) = 32 Then
        e.Handled = False
    Else
        e.Handled = True
    End If
End Sub
End Class
```

frmModificarEstudiantes

```
Public Class frmModificarEstudiantes
    Sub LlenarCombos()
    With Me.cboLugarNacimiento
        .Items.Add("Managua")
        .Items.Add("León")
        .Items.Add("Masaya")
        .Items.Add("Granada")
        .Items.Add("Corinto")
        .Items.Add("Chinandega")
        .Items.Add("El Viejo")
        .Items.Add("Rivas")
        .Items.Add("Telica")
    End With
    End Sub
End Class
```



```
.Items.Add("Quezalguaque")
.Items.Add("Boaco")
.Items.Add("Carazo")
.Items.Add("Chontales")
.Items.Add("Estelí")
.Items.Add("Jinotega")
.Items.Add("Matagalpa")
.Items.Add("Madriz")
.Items.Add("Nueva Segobia")
.Items.Add("Río San Juan")
.Items.Add("Atlántico Norte")
.Items.Add("Atlántico Sur")
.Items.Add("Otro")
End With
With Me.cboSexo
.Items.Add("M")
.Items.Add("F")
End With
With Me.cboTipoSangre
.Items.Add("A+")
.Items.Add("A-")
.Items.Add("O+")
.Items.Add("O-")
.Items.Add("AB+")
.Items.Add("AB-")
End With

With Me.mskTelefono
.Mask = ""
.Mask = "###-####"
End With
End Sub

Sub ModificarEstudiantes()
Dim CadenaConexion As String
Try
CadenaConexion = modUtilidades.Conexion
Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
Dim ocmd As SqlCommand = New
SqlCommand.SqlCommand("proc_ModificarEstudiante", cn)
Dim oParametros As New SqlParameter()
'Identificador
oParametros = ocmd.Parameters.Add("@IdEstudiante", SqlDbType.Char, 8)
oParametros.Value = Me.txtIdentificador.Text
```



```
'nombres
oParametros = ocmd.Parameters.Add("@Nombres", SqlDbType.Char, 250)
oParametros.Value = Me.txtNombres.Text
'apellidos
oParametros = ocmd.Parameters.Add("@Apellidos", SqlDbType.Char, 250)
oParametros.Value = Me.txtApellidos.Text
'lugar de nacimiento
oParametros = ocmd.Parameters.Add("@LugarNacimiento", SqlDbType.Char,
250)
oParametros.Value = Me.cboLugarNacimiento.Text
'fecha de nacimiento
oParametros = ocmd.Parameters.Add("@FechaNacimiento",
SqlDbType.DateTime)
oParametros.Value = Me.dtpFechaNacimiento.Text
'edad
oParametros = ocmd.Parameters.Add("@Edad", SqlDbType.Int)
oParametros.Value = Me.txtEdad.Text
'sexo
oParametros = ocmd.Parameters.Add("@Sexo", SqlDbType.Char, 1)
oParametros.Value = Me.cboSexo.Text
'tipo de sangre
oParametros = ocmd.Parameters.Add("@TipoSangre", SqlDbType.Char, 3)
oParametros.Value = Me.cboTipoSangre.Text
'dirección
oParametros = ocmd.Parameters.Add("@Direccion", SqlDbType.Char, 500)
oParametros.Value = Me.txtDireccion.Text
'teléfono
oParametros = ocmd.Parameters.Add("@Telefono", SqlDbType.Char, 8)
oParametros.Value = Me.mskTelefono.Text

ocmd.CommandType = CommandType.StoredProcedure
cn.Open()
Dim oda As New Data.SqlClient.SqlDataAdapter(ocmd)
Dim ods As New Data.DataSet()
oda.Fill(ods)
'Cerramos o liberamos recursos
ods.Dispose()
oda.Dispose()
ocmd.Dispose()
cn.Close()
cn.Dispose()
MsgBox("Estudiante modificado", MsgBoxStyle.Exclamation)
Catch ex As Exception
System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
```



```
End Try  
End Sub
```

```
Private Sub cmdCancelar_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles cmdCancelar.Click  
    Me.Hide()  
    Me.Close()  
    frmPrincipal.IsMdiContainer = True 'Se utiliza para especificar que el FrmPadre es  
un Mdi Padre  
    frmCatEstudiantes.MdiParent = frmPrincipal 'Indica que es un formulario  
secundario  
    frmCatEstudiantes.Show() 'Muestra el formulario  
End Sub
```

```
Private Sub cmdAceptar_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles cmdAceptar.Click  
    If Len(Me.txtNombres.Text) = 0 Then  
        MsgBox("Debe especificar los nombres del estudiante", MsgBoxStyle.Critical,  
"ISSS")  
        Me.txtNombres.Focus()  
        Exit Sub  
    ElseIf Len(Me.txtApellidos.Text) = 0 Then  
        MsgBox("Debe especificar los apellidos del estudiante", MsgBoxStyle.Critical,  
"ISSS")  
        Me.txtApellidos.Focus()  
        Exit Sub  
    ElseIf Len(Me.cboLugarNacimiento.Text) = 0 Then  
        MsgBox("Debe especificar el lugar de nacimiento del estudiante",  
MsgBoxStyle.Critical, "ISSS")  
        Me.cboLugarNacimiento.Focus()  
        Exit Sub  
    ElseIf Len(Me.txtEdad.Text) = 0 Then  
        MsgBox("Debe especificar la edad del estudiante", MsgBoxStyle.Critical, "ISSS")  
        Me.txtEdad.Focus()  
        Exit Sub  
    ElseIf Len(Me.cboSexo.Text) = 0 Then  
        MsgBox("Debe especificar el sexo del estudiante", MsgBoxStyle.Critical, "ISSS")  
        Me.cboSexo.Focus()  
        Exit Sub  
    ElseIf Len(Me.cboTipoSangre.Text) = 0 Then
```



```
MsgBox("Debe especificar el tipo de sangre del estudiante", MsgBoxStyle.Critical,
"ISSS")
    Me.cboTipoSangre.Focus()
    Exit Sub
    ElseIf Len(Me.txtDireccion.Text) = 0 Then
        MsgBox("Debe especificar la dirección del estudiante", MsgBoxStyle.Critical,
"ISSS")
        Me.txtDireccion.Focus()
        Exit Sub
        ElseIf Len(Me.mskTelefono.Text)= 0 Or Me.mskTelefono="" - "" Then
            MsgBox("Debe especificar el número de teléfono del estudiante",
MsgBoxStyle.Critical, "ISSS")
            Me.mskTelefono.Focus()
            Exit Sub
        Else
            Call Me.ModificarEstudiantes()
            Me.txtNombres.Clear()
            Me.txtApellidos.Clear()
            Me.txtEdad.Clear()
            Me.txtDireccion.Clear()
            Me.mskTelefono.Clear()
            Me.Hide()
            Me.Close()
            frmPrincipal.IsMdiContainer = True 'Se utiliza para especificar que el FrmPadre
es un Mdi Padre
            frmCatEstudiantes.MdiParent = frmPrincipal 'Indica que es un formulario
secundario
            frmCatEstudiantes.Show() 'Muestra el formulario
        End If
    End Sub

Private Sub txtNombres_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtNombres.KeyPress
    If (Asc(e.KeyChar)) >= 65 And (Asc(e.KeyChar)) <= 90 Or _
(Asc(e.KeyChar)) >= 97 And (Asc(e.KeyChar)) <= 122 Or _
(Asc(e.KeyChar)) = System.Windows.Forms.Keys.Back Or _
(Asc(e.KeyChar)) = 44 Or (Asc(e.KeyChar)) = 46 Or _
(Asc(e.KeyChar)) = 241 Or (Asc(e.KeyChar)) = 209 Or _
(Asc(e.KeyChar)) = 160 Or (Asc(e.KeyChar)) = 130 Or _
(Asc(e.KeyChar)) = 161 Or (Asc(e.KeyChar)) = 162 Or _
(Asc(e.KeyChar)) = 163 Or (Asc(e.KeyChar)) = 129 Or _
(Asc(e.KeyChar)) = 154 Or _
(Asc(e.KeyChar)) = 32 Then
        e.Handled = False
```



```
Else
    e.Handled = True
End If
End Sub
```

```
Private Sub txtApellidos_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtApellidos.KeyPress
    If (Asc(e.KeyChar)) >= 65 And (Asc(e.KeyChar)) <= 90 Or _
        (Asc(e.KeyChar)) >= 97 And (Asc(e.KeyChar)) <= 122 Or _
        (Asc(e.KeyChar)) = System.Windows.Forms.Keys.Back Or _
        (Asc(e.KeyChar)) = 44 Or (Asc(e.KeyChar)) = 46 Or _
        (Asc(e.KeyChar)) = 241 Or (Asc(e.KeyChar)) = 209 Or _
        (Asc(e.KeyChar)) = 160 Or (Asc(e.KeyChar)) = 130 Or _
        (Asc(e.KeyChar)) = 161 Or (Asc(e.KeyChar)) = 162 Or _
        (Asc(e.KeyChar)) = 163 Or (Asc(e.KeyChar)) = 129 Or _
        (Asc(e.KeyChar)) = 154 Or _
        (Asc(e.KeyChar)) = 32 Then
        e.Handled = False
    Else
        e.Handled = True
    End If
End Sub
```

```
Private Sub txtEdad_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtEdad.KeyPress
    If (Asc(e.KeyChar)) >= 48 And (Asc(e.KeyChar)) <= 57 Or _
        (Asc(e.KeyChar)) = System.Windows.Forms.Keys.Back Then
        e.Handled = False
    Else
        e.Handled = True
    End If
End Sub
```

```
Private Sub txtDireccion_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtDireccion.KeyPress
    If (Asc(e.KeyChar)) >= 65 And (Asc(e.KeyChar)) <= 90 Or _
        (Asc(e.KeyChar)) >= 97 And (Asc(e.KeyChar)) <= 122 Or _
        (Asc(e.KeyChar)) = System.Windows.Forms.Keys.Back Or _
        (Asc(e.KeyChar)) = 44 Or (Asc(e.KeyChar)) = 46 Or _
        (Asc(e.KeyChar)) = 241 Or (Asc(e.KeyChar)) = 209 Or _
        (Asc(e.KeyChar)) = 160 Or (Asc(e.KeyChar)) = 130 Or _
        (Asc(e.KeyChar)) = 161 Or (Asc(e.KeyChar)) = 162 Or _
```



```
(Asc(e.KeyChar)) = 163 Or (Asc(e.KeyChar)) = 129 Or _  
(Asc(e.KeyChar)) >= 48 And (Asc(e.KeyChar)) <= 57 Or _  
(Asc(e.KeyChar)) = 154 Or _  
(Asc(e.KeyChar)) = 47 Or _  
(Asc(e.KeyChar)) = 32 Then  
    e.Handled = False  
Else  
    e.Handled = True  
End If  
End Sub
```

```
Private Sub frmModificarEstudiantes_Load(ByVal sender As Object, ByVal e As  
System.EventArgs) Handles Me.Load  
    Call Me.LlenarCombos()  
End Sub  
End Class
```

frmCatAulas

```
Public Class frmCatAulas  
    Sub CrearListView()  
        'Formatear el control ListView  
        With Me.lvAulas  
            .Clear()  
            .LabelEdit = False  
            .AllowColumnReorder = True  
            .FullRowSelect = True  
            .Sorting = SortOrder.Ascending  
            .View = View.Details  
            .GridLines = True  
            .Columns.Add("Id", 60, HorizontalAlignment.Center)  
            .Columns.Add("Nombre", 200, HorizontalAlignment.Center)  
            .Columns.Add("Descripción", 250, HorizontalAlignment.Center)  
        End With  
    End Sub  
  
    Sub CargarAulas()  
        Dim CadenaConexion As String  
        CadenaConexion = modUtilidades.Conexion  
        Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)  
        Dim vListarCargos As New Data.SqlClient.SqlCommand("proc_ListarAulas", cn)  
        vListarCargos.CommandType = CommandType.StoredProcedure  
        ' En la variable drListarCargos se almacenaran los datos  
        ' devueltos por la ejecución del procedimiento
```



```
' almacenado proc_ListarAulas
Dim drListarCargos As Data.SqlClient.SqlDataReader
cn.Open()
drListarCargos = vListarCargos.ExecuteReader
'Llenar el ListView
Dim vpos As Integer
Me.lvAulas.Items.Clear()
While drListarCargos.Read
    With Me.lvAulas
        .Items.Add(drListarCargos("idAula")) ' Identificador
        .Items(vpos).SubItems.Add(drListarCargos("Nombre")) ' Nombre
        .Items(vpos).SubItems.Add(drListarCargos("Descripcion")) ' Descripcion
    ' Llenar el combo con los nombres de las aulas
    Me.cboNombreAulas.Items.Add(Trim(drListarCargos("Nombre"))) ' Nombre
    vpos += 1
    End With
End While
If Me.cboNombreAulas.Items.Count = 0 Then
    Me.cmdBorrar.Enabled = False
    Me.cmdModificar.Enabled = False
    Me.cboNombreAulas.Enabled = False
    Me.cmdNueva.Focus()
Else
    Me.cmdBorrar.Enabled = True
    Me.cmdModificar.Enabled = True
    Me.cboNombreAulas.Enabled = True
    Me.cboNombreAulas.SelectedIndex = 0
    Me.cmdNueva.Focus()
End If
'Cerrando objetos
drListarCargos.Close()
cn.Close()
```

End Sub

```
Sub AulasXNombre()
    Dim CadenaConexion As String
    Dim f As Integer
    Try
        CadenaConexion = modUtilidades.Conexion
        Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
        Dim ocmd As SqlCommand = New
        SqlCommand("proc_AulasXNombre", cn)
```



```
Dim oParametros As New SqlClient.SqlParameter()  
'nombre del aula a buscar  
If Me.cboNombreAulas.Text = "" Then  
    Call Me.CargarAulas()  
Else  
    oParametros = ocmd.Parameters.Add("@nombre", SqlDbType.Char, 250)  
    oParametros.Value = Me.cboNombreAulas.Text  
    ocmd.CommandType = CommandType.StoredProcedure  
    cn.Open()  
    Dim oda As New Data.SqlClient.SqlDataAdapter(ocmd)  
    Dim odt As New DataTable()  
    oda.Fill(odt)  
    ' Creamos un objeto DataRow y nos ubicamos en el 1er registro devuelto  
    Dim DatosTabla As Data.DataRow  
    If odt.Rows.Count = 0 Then  
        Exit Sub  
    Else  
        f = odt.Rows.Count - 1  
        DatosTabla = odt.Rows(f)  
        Call Me.CrearListView()  
        f = 0  
        While f <= odt.Rows.Count - 1  
            DatosTabla = odt.Rows(f) ' nos movemos por los regs resultantes  
            Me.lvAulas.Items.Add(DatosTabla("idAula"))  
            Me.lvAulas.Items(f).SubItems.Add(DatosTabla("Nombre"))  
            Me.lvAulas.Items(f).SubItems.Add(DatosTabla("Descripcion"))  
            f += 1  
        End While  
        odt.Dispose()  
        oda.Dispose()  
        ocmd.Dispose()  
        cn.Close()  
        cn.Dispose()  
  
        End If  
    End If  
    Catch ex As Exception  
        System.Windows.Forms.MessageBox.Show(ex.Message.ToString)  
    End Try  
End Sub  
  
Private Sub frmCatAulas_Load(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MyBase.Load  
    Call Me.CrearListView()
```



```
Call Me.CargarAulas()  
End Sub
```

```
Private Sub cmdCancelar_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles cmdCancelar.Click  
    Me.Close()  
End Sub
```

```
Private Sub cboNombreAulas_KeyUp(ByVal sender As Object, ByVal e As  
System.Windows.Forms.KeyEventArgs) Handles cboNombreAulas.KeyUp  
    Call Me.AulasXNombre()  
End Sub
```

```
Private Sub cmdModificar_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles cmdModificar.Click  
    Dim FilaActual As ListViewItem  
    Dim IdAux As String  
    Dim NombreAux As String  
    Dim DescripcionAux As String  
    Try  
        If Me.lvAulas.SelectedItems.Count = 0 Then  
            MsgBox("Debe especificar el aula a modificar", MsgBoxStyle.Exclamation)  
            Exit Sub  
        Else  
            FilaActual = Me.lvAulas.SelectedItems(0)  
            IdAux = FilaActual.SubItems(0).Text  
            NombreAux = Trim(FilaActual.SubItems(1).Text)  
            DescripcionAux = Trim(FilaActual.SubItems(2).Text)  
            With frmModificarAula  
                .txtIdentificador.Text = IdAux  
                .txtNombre.Text = NombreAux  
                .txtDescripcion.Text = DescripcionAux  
                Me.Hide()  
                Me.Close()  
                .Show()  
            End With  
        End If  
    Catch ex As Exception  
        System.Windows.Forms.MessageBox.Show(ex.Message.ToString)  
    End Try  
End Sub  
  
Sub EliminarAula()
```



```
Dim IdAux As String
Dim CadenaConexion As String
Dim FilaActual As ListViewItem
Try
    'Obtener el Identificador del aula a borrar
    FilaActual = Me.lvAulas.SelectedItem(0)
    IdAux = FilaActual.SubItems(0).Text
    CadenaConexion = modUtilidades.Conexion
    Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
    Dim ocmd As SqlCommand = New
    SqlCommand("proc_EliminarAula", cn)
    Dim oParametros As New SqlCommandParameter()
    'Identificador
    oParametros = ocmd.Parameters.Add("@idAula", SqlDbType.Char, 4)
    oParametros.Value = IdAux
    ocmd.CommandType = CommandType.StoredProcedure
    cn.Open()
    Dim oda As New Data.SqlClient.SqlDataAdapter(ocmd)
    Dim ods As New DataSet()
    oda.Fill(ods)
    'Cerramos o liberamos recursos
    ods.Dispose()
    oda.Dispose()
    ocmd.Dispose()
    cn.Close()
    cn.Dispose()
    MsgBox("Aula eliminada")
Catch ex As Exception
    System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub

Private Sub cmdBorrar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdBorrar.Click
    Try

        If Me.lvAulas.SelectedItem.Count = 0 Then
            MsgBox("Debe especificar el aula a borrar", MsgBoxStyle.Exclamation)
            Exit Sub
        Else
            Call Me.EliminarAula()
            Call Me.CargarAulas()
        End If
    Catch ex As Exception
```



```
System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub
```

```
Private Sub cmdNueva_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdNueva.Click
    Me.Hide()
    Me.Close()
    frmAula.Show()
End Sub
End Class
```

frmNuevaAula

```
Public Class frmAula
    Sub UltimoCodigoAula()
        Dim CadenaConexion As String
        CadenaConexion = modUtilidades.Conexion
        Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
        Dim vUltimoCodigo As New Data.SqlClient.SqlCommand("proc_GenerarIdAula",
cn)
        vUltimoCodigo.CommandType = CommandType.StoredProcedure
        Dim drUltimoCodigo As Data.SqlClient.SqlDataReader
        cn.Open()
        drUltimoCodigo = vUltimoCodigo.ExecuteReader
        If drUltimoCodigo.Read Then
            Me.txtIdentificador.Text = Format(drUltimoCodigo("MaxCodigo"), "0000")
        Else
            Me.txtIdentificador.Text = Format("1", "0000")
        End If
        drUltimoCodigo.Close()
        cn.Close()
    End Sub

    Sub InsertarNuevaAula()
        Dim CadenaConexion As String
        CadenaConexion = modUtilidades.Conexion

        Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
        Dim cmd As SqlCommand = New
SqlCommand("proc_InsertarAula", cn)
        Dim Parametros As New SqlParameter()
'codigo
        Parametros = cmd.Parameters.Add("@id", SqlDbType.Char, 4)
        Parametros.Value = Me.txtIdentificador.Text
    End Sub
End Class
```



'nombre

Parametros = cmd.Parameters.Add("@nombre", SqlDbType.Char, 250)

Parametros.Value = Me.txtNombre.Text

'descripcion

Parametros = cmd.Parameters.Add("@descripcion", SqlDbType.Char, 250)

Parametros.Value = Me.txtDescripcion.Text

cmd.CommandType = CommandType.StoredProcedure

cn.Open()

Dim da As New Data.SqlClient.SqlDataAdapter(cmd)

Dim ds As New Data.DataSet()

da.Fill(ds)

'Cerramos o liberamos recursos

ds.Dispose()

da.Dispose()

cmd.Dispose()

cn.Close()

cn.Dispose()

End Sub

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

Call Me.UltimoCodigoAula()

End Sub

Private Sub cmdCancelar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdCancelar.Click

Me.Hide()

Me.Close()

frmPrincipal.IsMdiContainer = True 'Se utiliza para especificar que el FrmPadre es un Mdi Padre

frmCatAulas.MdiParent = frmPrincipal 'Indica que es un formulario secundario

frmCatAulas.Show() 'Muestra el formulario

End Sub

Private Sub cmdAceptar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdAceptar.Click

If Len(Me.txtNombre.Text) = 0 Then

MsgBox("Debe especificar el nombre del aula", MsgBoxStyle.Critical, "ISSS")

Me.txtNombre.Focus()

Exit Sub

Elseif Len(Me.txtDescripcion.Text) = 0 Then

MsgBox("Debe especificar la descripción del aula", MsgBoxStyle.Critical, "ISSS")

Me.txtDescripcion.Focus()

Exit Sub



Else

Call Me.InsertarNuevaAula()

Me.txtNombre.Clear()

Me.txtDescripcion.Clear()

Me.Hide()

Me.Close()

frmPrincipal.IsMdiContainer = True 'Se utiliza para especificar que el

FrmPadre es un Mdi Padre

frmCatAulas.MdiParent = frmPrincipal 'Indica que es un formulario secundario

frmCatAulas.Show() 'Muestra el formulario

End If

End Sub

Private Sub txtNombre_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtNombre.KeyPress

If (Asc(e.KeyChar)) >= 65 And (Asc(e.KeyChar)) <= 90 Or _

(Asc(e.KeyChar)) >= 97 And (Asc(e.KeyChar)) <= 122 Or _

(Asc(e.KeyChar)) = System.Windows.Forms.Keys.Back Or _

(Asc(e.KeyChar)) = 44 Or (Asc(e.KeyChar)) = 46 Or _

(Asc(e.KeyChar)) = 241 Or (Asc(e.KeyChar)) = 209 Or _

(Asc(e.KeyChar)) = 160 Or (Asc(e.KeyChar)) = 130 Or _

(Asc(e.KeyChar)) = 161 Or (Asc(e.KeyChar)) = 162 Or _

(Asc(e.KeyChar)) = 163 Or (Asc(e.KeyChar)) = 129 Or _

(Asc(e.KeyChar)) = 154 Or _

(Asc(e.KeyChar)) = 32 Then

e.Handled = False

Else

e.Handled = True

End If

End Sub

Private Sub txtDescripcion_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtDescripcion.KeyPress

If (Asc(e.KeyChar)) >= 65 And (Asc(e.KeyChar)) <= 90 Or _

(Asc(e.KeyChar)) >= 97 And (Asc(e.KeyChar)) <= 122 Or _

(Asc(e.KeyChar)) = System.Windows.Forms.Keys.Back Or _

(Asc(e.KeyChar)) = 44 Or (Asc(e.KeyChar)) = 46 Or _

(Asc(e.KeyChar)) = 241 Or (Asc(e.KeyChar)) = 209 Or _

(Asc(e.KeyChar)) = 160 Or (Asc(e.KeyChar)) = 130 Or _

(Asc(e.KeyChar)) = 161 Or (Asc(e.KeyChar)) = 162 Or _

(Asc(e.KeyChar)) = 163 Or (Asc(e.KeyChar)) = 129 Or _

(Asc(e.KeyChar)) = 154 Or _

(Asc(e.KeyChar)) = 32 Then



```
e.Handled = False
Else
e.Handled = True
End If
End Sub
End Class
```

frmModificarAula

```
Public Class frmModificarAula
Sub ModificarAulas()
Dim CadenaConexion As String
Try
CadenaConexion = modUtilidades.Conexion
Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
Dim ocmd As SqlCommand = New
SqlCommand("proc_ModificarAulas", cn)
Dim oParametros As New SqlParameter()
'Identificador
oParametros = ocmd.Parameters.Add("@idAula", SqlDbType.Char, 4)
oParametros.Value = Me.txtIdentificador.Text
'nombre
oParametros = ocmd.Parameters.Add("@nombre", SqlDbType.Char, 250)
oParametros.Value = Me.txtNombre.Text
'descripcion
oParametros = ocmd.Parameters.Add("@descripcion", SqlDbType.Char, 250)
oParametros.Value = Me.txtDescripcion.Text
ocmd.CommandType = CommandType.StoredProcedure
cn.Open()
Dim oda As New Data.SqlClient.SqlDataAdapter(ocmd)
Dim ods As New Data.DataSet()
oda.Fill(ods)
'Cerramos o liberamos recursos
ods.Dispose()
oda.Dispose()
ocmd.Dispose()

cn.Close()
cn.Dispose()
MsgBox("Aula modificada", MsgBoxStyle.Exclamation)
Catch ex As Exception
System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub
```



```
Private Sub cmdCancelar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdCancelar.Click
    Me.Hide()
    Me.Close()
    frmPrincipal.IsMdiContainer = True 'Se utiliza para especificar que el FrmPadre es un Mdi Padre
    frmCatAulas.MdiParent = frmPrincipal 'Indica que es un formulario secundario
    frmCatAulas.Show() 'Muestra el formulario
End Sub
```

```
Private Sub cmdAceptar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdAceptar.Click
    If Len(Me.txtNombre.Text) = 0 Then
        MsgBox("Debe especificar el nombre del aula", MsgBoxStyle.Critical, "INSS")
        Me.txtNombre.Focus()
        Exit Sub
    ElseIf Len(Me.txtDescripcion.Text) = 0 Then
        MsgBox("Debe especificar la descripción del aula", MsgBoxStyle.Critical, "INSS")
        Me.txtDescripcion.Focus()
        Exit Sub
    Else
        Call Me.ModificarAulas()
        Me.Hide()
        Me.Close()
        frmPrincipal.IsMdiContainer = True 'Se utiliza para especificar que el FrmPadre es un Mdi Padre
        frmCatAulas.MdiParent = frmPrincipal 'Indica que es un formulario secundario
        frmCatAulas.Show() 'Muestra el formulario
    End If
End Sub
```

```
Private Sub txtNombre_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtNombre.KeyPress
    If (Asc(e.KeyChar)) >= 65 And (Asc(e.KeyChar)) <= 90 Or _
        (Asc(e.KeyChar)) >= 97 And (Asc(e.KeyChar)) <= 122 Or _
        (Asc(e.KeyChar)) = System.Windows.Forms.Keys.Back Or _
        (Asc(e.KeyChar)) = 44 Or (Asc(e.KeyChar)) = 46 Or _
        (Asc(e.KeyChar)) = 241 Or (Asc(e.KeyChar)) = 209 Or _
        (Asc(e.KeyChar)) = 160 Or (Asc(e.KeyChar)) = 130 Or _
        (Asc(e.KeyChar)) = 161 Or (Asc(e.KeyChar)) = 162 Or _
        (Asc(e.KeyChar)) = 163 Or (Asc(e.KeyChar)) = 129 Or _
        (Asc(e.KeyChar)) = 154 Or _
        (Asc(e.KeyChar)) = 32 Then
```



```
e.Handled = False
Else
e.Handled = True
End If
End Sub
```

```
Private Sub txtDescripcion_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtDescripcion.KeyPress
If (Asc(e.KeyChar)) >= 65 And (Asc(e.KeyChar)) <= 90 Or _
(Asc(e.KeyChar)) >= 97 And (Asc(e.KeyChar)) <= 122 Or _
(Asc(e.KeyChar)) = System.Windows.Forms.Keys.Back Or _
(Asc(e.KeyChar)) = 44 Or (Asc(e.KeyChar)) = 46 Or _
(Asc(e.KeyChar)) = 241 Or (Asc(e.KeyChar)) = 209 Or _
(Asc(e.KeyChar)) = 160 Or (Asc(e.KeyChar)) = 130 Or _
(Asc(e.KeyChar)) = 161 Or (Asc(e.KeyChar)) = 162 Or _
(Asc(e.KeyChar)) = 163 Or (Asc(e.KeyChar)) = 129 Or _
(Asc(e.KeyChar)) = 154 Or _
(Asc(e.KeyChar)) = 32 Then
e.Handled = False
Else
e.Handled = True
End If
End Sub
End Class
```

frmRecibosXFecha

```
Public Class frmRptRecibosXFecha
' Esta variable la usaré para pasarle el parámetro idRecibo al reporte rptRecibos
Friend IdReciboAux As String
Sub CrearListViewRecibo()
With Me.lvRecibos
.Clear()
.LabelEdit = False
.AllowColumnReorder = True
.FullRowSelect = True
.Sorting = SortOrder.Ascending

.View = View.Details
.GridLines = True
.Columns.Add("Recibo", 65, HorizontalAlignment.Center)
.Columns.Add("Estudiante", 70, HorizontalAlignment.Center)
.Columns.Add("Referencia", 75, HorizontalAlignment.Center)
.Columns.Add("Fecha", 80, HorizontalAlignment.Center)

```



```
.Columns.Add("Concepto", 200, HorizontalAlignment.Center)
.Columns.Add("Total", 70, HorizontalAlignment.Center)
.Columns.Add("Valor en letras", 100, HorizontalAlignment.Center)
```

End With

End Sub

Sub CrearListViewDetalleR()

With Me.lvDetalleRecibo

```
.Clear()
.LabelEdit = False
.AllowColumnReorder = True
.FullRowSelect = True
.Sorting = SortOrder.Ascending
.View = View.Details
.GridLines = True
.Columns.Add("Id. Recibo", 75, HorizontalAlignment.Center)
.Columns.Add("Id. Arancel", 75, HorizontalAlignment.Center)
.Columns.Add("Descripción", 230, HorizontalAlignment.Center)
.Columns.Add("Año pagado", 75, HorizontalAlignment.Center)
.Columns.Add("Mes pagado", 75, HorizontalAlignment.Center)
.Columns.Add("Sub Total", 80, HorizontalAlignment.Center)
```

End With

End Sub

Sub RecibosXFecha()

Dim CadenaConexion As String

Dim f As Integer

Try

```
CadenaConexion = modUtilidades.Conexion
Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
Dim cmd As SqlCommand = New
SqlCommand("proc_RptListarRecibosXFecha", cn)
Dim Parametros As New SqlParameter()
'Fecha
Parametros = cmd.Parameters.Add("@Fecha", SqlDbType.DateTime)
Parametros.Value = Mid(Me.dtpFecha.Value, 1, 10)
cmd.CommandType = CommandType.StoredProcedure
cn.Open()
```

```
Dim da As New SqlDataAdapter(cmd)
```

```
Dim dt As New DataTable()
```

```
da.Fill(dt)
```

```
' Creamos un objeto DataRow y nos ubicamos en el 1er registro devuelto
```

```
Dim DatosTabla As DataRow
```



```
If dt.Rows.Count = 0 Then
    Call Me.CrearListViewRecibo()
    Exit Sub
Else
    Call Me.CrearListViewRecibo()
    f = 0
    While f <= dt.Rows.Count - 1
        DatosTabla = dt.Rows(f) ' nos movemos por los regs resultantes
        Me.lvRecibos.Items.Add(DatosTabla("idRecibo"))
        Me.lvRecibos.Items(f).SubItems.Add(DatosTabla("idEstudiante"))
        Me.lvRecibos.Items(f).SubItems.Add(DatosTabla("Referencia"))
        Me.lvRecibos.Items(f).SubItems.Add(DatosTabla("FechaRecibo"))
        Me.lvRecibos.Items(f).SubItems.Add(DatosTabla("Concepto"))
        Me.lvRecibos.Items(f).SubItems.Add(DatosTabla("Total"))
        Me.lvRecibos.Items(f).SubItems.Add(DatosTabla("ValorEnLetras"))
        f += 1
    End While
    dt.Dispose()
    da.Dispose()
    cmd.Dispose()
    cn.Close()
    cn.Dispose()
End If
Catch ex As Exception
    System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub

Private Sub cmdBuscarRecibos_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdBuscarRecibos.Click
    Call Me.RecibosXFecha()
End Sub

Private Sub cmdCerrar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdCerrar.Click
    Me.Hide()
    Me.Close()
End Sub

Private Sub mskFecha_ValueChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles dtpFecha.ValueChanged
    Call Me.RecibosXFecha()
End Sub
```



```
Private Sub lvRecibos_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles lvRecibos.Click
    Dim FilaActual As ListViewItem
    Dim IdRecibo As String
    Dim CadenaConexion As String
    Dim f As Integer
    Try
        FilaActual = Me.lvRecibos.SelectedItems(0)
        IdRecibo = FilaActual.SubItems(0).Text
        ' Le asigno el valor a la variable IdReciboAux
        ' para mostrar el reporte rptRecibos.rpt
        IdReciboAux = IdRecibo
        CadenaConexion = modUtilidades.Conexion
        Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
        Dim cmd As SqlCommand = New
        SqlCommand.SqlCommand("proc_ListarRecibosXIdRecibo", cn)
        Dim Parametros As New SqlParameter()
        'Identificador del recibo
        Parametros = cmd.Parameters.Add("@IdRecibo", SqlDbType.Char, 8)
        Parametros.Value = IdRecibo
        cmd.CommandType = CommandType.StoredProcedure
        cn.Open()
        Dim da As New Data.SqlClient.SqlDataAdapter(cmd)
        Dim dt As New DataTable()
        da.Fill(dt)
        ' Creo un objeto DataRow y nos ubicamos en el 1er registro devuelto
        Dim DatosTabla As DataRow
        If dt.Rows.Count = 0 Then
            Call Me.CrearListViewDetalleR()
            Exit Sub
        Else
            f = dt.Rows.Count - 1
            DatosTabla = dt.Rows(f)
            Call Me.CrearListViewDetalleR()
            f = 0
            While f <= dt.Rows.Count - 1
                DatosTabla = dt.Rows(f) ' nos movemos por los regs resultantes
                Me.lvDetalleRecibo.Items.Add(DatosTabla("idRecibo"))
                Me.lvDetalleRecibo.Items(f).SubItems.Add(DatosTabla("idArancel"))

                Me.lvDetalleRecibo.Items(f).SubItems.Add(DatosTabla("Descripcion"))
                Me.lvDetalleRecibo.Items(f).SubItems.Add(DatosTabla("AnyoPagado"))
                Me.lvDetalleRecibo.Items(f).SubItems.Add(DatosTabla("MesPagado"))
                Me.lvDetalleRecibo.Items(f).SubItems.Add(DatosTabla("SubTotal"))
            End While
        End If
    Catch
    End Try
End Sub
```



```
        f += 1
    End While
    dt.Dispose()
    da.Dispose()
    cmd.Dispose()
    cn.Close()
    cn.Dispose()
End If
Catch ex As Exception
    System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub

Private Sub frmRptRecibosXFecha_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
    Call Me.CrearListViewRecibo()
    Call Me.CrearListViewDetalleR()
End Sub

Private Sub cmdImprimir_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdImprimir.Click
    frmMostrarRecibo.Show()
End Sub
End Class
```

Reporte comprobante de ingreso a caja (reporte de recibo)

```
Public Class frmMostrarRecibo
    Friend crApp As New CRAXDRT.Application
    Friend crReport As New CRAXDRT.Report
    Public mflgContinuar As Boolean
    Public mstrParametro As String
    Private Sub frmMostrarRecibo_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Dim crParamDefs As CRAXDRT.ParameterFieldDefinitions
        Dim crParamDef As CRAXDRT.ParameterFieldDefinition
        mflgContinuar = True
        crReport = crApp.OpenReport(modUtilidades.RutaReportes + "rptRecibos.rpt")
        crParamDefs = crReport.ParameterFields
        For Each crParamDef In crParamDefs

            Select Case crParamDef.ParameterFieldName
                Case "@IdRecibo"
                    crParamDef.AddDefaultValue(frmRptRecibosXFecha.IdReciboAux)
```



```
        crParamDef.AddCurrentValue(frmRptRecibosXFecha.IdReciboAux)
    End Select
Next
Me.AxCrViewer1.ReportSource = crReport
Me.AxCrViewer1.DisplayGroupTree = False
Me.AxCrViewer1.ViewReport()
Me.AxCrViewer1.Zoom(95)
crParamDefs = Nothing
crParamDef = Nothing
Exit Sub
ErrorHandler:
If Err.Number = -2147206461 Then
    MsgBox("El archivo de reporte no se encuentra, restáurelo de los discos de
instalación", vbCritical + vbOKOnly)
Else
    MsgBox(Err.Description + " " + Str(Err.Number), vbCritical + vbOKOnly)
End If
mflgContinuar = False
End Sub

Private Sub frmMostrarRecibo_Resize(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Resize
    Me.AxCrViewer1.Top = 0
    Me.AxCrViewer1.Left = 0
    Me.AxCrViewer1.Height = Me.Height
    Me.AxCrViewer1.Width = Me.Width
End Sub
End Class
```

frmEstadoDeCuenta

```
Public Class frmEstadoDeCuenta
    Sub MatriculasXAnyo()
        Dim CadenaConexion As String
        Dim f As Integer
        Try
            CadenaConexion = modUtilidades.Conexion
            Dim cn As New Data.SqlClient.SqlConnection(CadenaConexion)
            Dim ocmd As SqlCommand = New
SqlCommand.SqlCommand("proc_RptListarMatriculasXAnyo", cn)
            Dim oParametros As New SqlCommand.SqlParameter()

            'Año de la matrícula
            oParametros = ocmd.Parameters.Add("@Anyo", SqlDbType.Int)
```



```
oParametros.Value = Trim(Me.txtAnyo.Text)
ocmd.CommandType = CommandType.StoredProcedure
cn.Open()
Dim oda As New Data.SqlClient.SqlDataAdapter(ocmd)
Dim odt As New DataTable()
oda.Fill(odt)
' Verificamos si hay elementos en DatosTabla
Dim DatosTabla As Data.DataRow
If odt.Rows.Count = 0 Then
    MsgBox("No hay matrículas registradas en ese año",
MsgBoxStyle.Exclamation)
    Me.cmdImprimir.Visible = False
    Me.cboldEstudiantes.Text = ""
    Me.cboldEstudiantes.Items.Clear()
    Me.txtAnyo.Focus()
    Exit Sub
Else
    Me.cmdImprimir.Visible = True
    f = 0
    While f <= odt.Rows.Count - 1
        DatosTabla = odt.Rows(f) ' nos movemos por los regs resultantes
        Me.cboldEstudiantes.Items.Add(DatosTabla("idEstudiante"))
        f += 1
    End While
    Me.cboldEstudiantes.SelectedIndex = 0
    odt.Dispose()
    oda.Dispose()
    ocmd.Dispose()
    cn.Close()
    cn.Dispose()
End If
Catch ex As Exception
    System.Windows.Forms.MessageBox.Show(ex.Message.ToString)
End Try
End Sub

Private Sub cmdCancelar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdCancelar.Click
    Me.Hide()
    Me.Close()
End Sub
```



```
Private Sub cmdImprimir_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdImprimir.Click
    If Len(Me.txtAnyo.Text) = 0 Then
        MsgBox("Debe especificar el Año")
        Me.txtAnyo.Focus()
        Exit Sub
    ElseIf Len(Me.cboldEstudiantes.Text) = 0 Then
        MsgBox("Debe especificar el Id. Estudiante")
        Me.cboldEstudiantes.Focus()
        Exit Sub
    Else
        frmRptMostrarEstadoDeCuenta.Show()
    End If
End Sub

Private Sub frmEstadoDeCuenta_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    Me.txtAnyo.Text = Mid(DateString, 7, 4)
End Sub

Private Sub cmdListar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdListar.Click
    If Me.txtAnyo.Text = "" Then
        MsgBox("Debe especificar el Año de la matrícula")
        Me.txtAnyo.Focus()
        Exit Sub
    Else
        Call Me.MatriculasXAnyo()
    End If
End Sub
End Class
```

Reporte Estado de Cuenta

```
Public Class frmRptMostrarEstadoDeCuenta
    Friend crApp As New CRAXDRT.Application
    Friend crReport As New CRAXDRT.Report
    Public mflgContinuar As Boolean
    Public mstrParametro1 As Integer
    Public mstrParametro2 As String
    Private Sub frmRptMostrarEstadoDeCuenta_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        Dim crParamDefs As CRAXDRT.ParameterFieldDefinitions
```



```
Dim crParamDef As CRAXDRT.ParameterFieldDefinition
mflgContinuar = True
crReport = crApp.OpenReport(modUtilidades.RutaReportes +
"rptEstadoDeCuenta.rpt")
crParamDefs = crReport.ParameterFields
For Each crParamDef In crParamDefs
    Select Case crParamDef.ParameterFieldName
        Case "@Anyo"
crParamDef.AddDefaultValue(CInt(Trim(frmEstadoDeCuenta.txtAnyo.Text)))

crParamDef.AddCurrentValue(CInt(Trim(frmEstadoDeCuenta.txtAnyo.Text)))
        Case "@idEstudiante"
            crParamDef.AddDefaultValue(frmEstadoDeCuenta.cboldEstudiantes.Text)
            crParamDef.AddCurrentValue(frmEstadoDeCuenta.cboldEstudiantes.Text)
    End Select
Next
Me.AxCRViewer1.ReportSource = crReport
Me.AxCRViewer1.DisplayGroupTree = False
Me.AxCRViewer1.ViewReport()
Me.AxCRViewer1.Zoom(95)
crParamDefs = Nothing
crParamDef = Nothing
Exit Sub

ErrorHandler:
If Err.Number = -2147206461 Then
    MsgBox("El archivo de reporte no se encuentra, restáurelo de los discos de
instalación", vbCritical + vbOKOnly)
Else
    MsgBox(Err.Description + " " + Str(Err.Number), vbCritical + vbOKOnly)
End If
mflgContinuar = False
End Sub

Private Sub frmRptMostrarEstadoDeCuenta_Resize(ByVal sender As Object, ByVal e
As System.EventArgs) Handles Me.Resize
    Me.AxCRViewer1.Top = 0
    Me.AxCRViewer1.Left = 0
    Me.AxCRViewer1.Height = Me.Height

    Me.AxCRViewer1.Width = Me.Width
End Sub
End Class
```