

Universidad Nacional Autónoma de Nicaragua

UNAN-León

Facultad de Ciencias y Tecnología

Departamento de Computación.



Tesis para optar al título de Ingeniería en Sistema de  
Información.

Tema: Automatización del Registro Académico del Colegio  
Apostólico Lirio de los Valles de la Ciudad de León.

Integrantes:

- ✓ Br. Nelly Albertina Artola García.
- ✓ Br. Shirley Maley Calderón Romero.
- ✓ Br. Yendri Yacenia Iglesias Gutiérrez.

Tutor:

- ✓ Ing. Julio César González Moreno.

León, Nicaragua 21 de Octubre del 2010.





## AGRADECIMIENTO

Le agradecemos primeramente a *Dios*, por darnos la vida y por permitirnos llegar a culminar nuestro trabajo.

A nuestros *padres*, que tuvieron que limitarse muchas veces, para que nosotros lográramos lo que ahora le estamos agradeciendo con todo el corazón, ya que sus sueños y nuestra meta se están cumpliendo después de haber concluido un largo tiempo de esfuerzo, empeño y dedicación. Sin ellos no podríamos haber llegado hasta aquí.

A nuestro *tutor* **Ing. Julio César González Moreno**, por su esfuerzo, dedicación. Sus conocimientos, sus orientaciones, su manera de trabajar, su persistencia, su paciencia y su motivación han sido fundamentales. Le agradecemos por habernos ayudado a culminar nuestra carrera.

A nuestro *amigo* **José Gabriel Bárcenas**, por su tiempo y apoyo.

A nuestros *Profesores* ya que nos brindaron sus conocimientos, ayuda y paciencia en lo largo de este camino universitario. En especial al **profesor: Ing. Denis Leopoldo Espinoza**.

A nuestros *familiares* y *amigos* por su apoyo y motivación para que saliéramos adelante.

Y a todos aquellos que de una u otra manera nos desearon éxitos en nuestra tesis.

**Nelly Albertina Artola García.**  
**Shirley Maley Calderón Romero.**  
**Yendri Yacenia Iglesias Gutiérrez.**





## DEDICATORIA

**Dedico este trabajo con mucho amor primeramente a DIOS**, porque me ha dado el conocimiento, fuerza, salud, para ver completadas mis metas y por haberme regalado a dos personas maravillosas.

**Estas personas maravillosas que son mis Padres Ing. Natividad Artola González y Lic. Albertina del Carmen García Aguilar**, siempre me han brindado todo su apoyo, comprensión y porque con mucho sacrificio me han sabido llevar por el buen camino. Me han enseñado a encarar las adversidades sin perder nunca la dignidad ni desfallecer en el intento. Me han dado todo lo que soy como persona, mis valores, mis principios, mi perseverancia, mi empeño y todo ello con una gran dosis de amor y sin pedir nunca nada a cambio. Los amo papá y mamá.

**A mis hermanos** Por darme su apoyo incondicional. Los quiero mucho.

**A Manuel de Jesús Ramírez Mayorga**, Por brindarme su ayuda desinteresadamente, esfuerzo y apoyo en momentos de decline y cansancio.

**A mis Compañeras Shirley Maley Calderón Romero y Yendri Yacenia Iglesias Gutiérrez** que compartieron todos sus conocimientos conmigo y ayudaron a la culminación de nuestra monografía.

**A nuestro Tutor Ing. Julio César González Moreno** por su esfuerzo, dedicación y su motivación han sido fundamentales.

Y a todas las personas que nos desearon éxito para culminar este trabajo.

**Br. Nelly Albertina Artola García.**





## DEDICATORIA

A **Dios** porque siempre ha estado conmigo en cada paso de mi vida, guiándome cuidándome y dándome fortaleza para continuar.

A **mis Padres: Leyla Romero y Mauricio Calderón** con mucho respeto amor y cariño que me brindaron siempre su apoyo económico e incondicional y que con mucho sacrificio me han sabido llevar por el buen camino y me han regalado siempre todo lo mejor, sobretodo mis estudios.

A **mi Madrina: Felicita Cruz Sandoval** que siempre me ha brindado todo lo mejor y he contado con ella en todo momento.

A **mi Mama Lidia Rocha (q.e.p.d)** mi abuela que en vida nos brindo su alegría y amor y me enseñó que se debe tener valor y resignación para enfrentar a la muerte. Que Dios la tenga con bien.

A **mis Compañeras** y amigas de Tesis que me tuvieron la paciencia más grande del mundo.

A **nuestro Tutor y amigo Ing. Julio César González Moreno** que nos brindo su tiempo y esmero en la realización de nuestro trabajo.

A todas las personas que de una u otra manera pusieron su confianza en mí y me brindaron siempre su apoyo.

**“El que Persevera Alcanza y Quien Verdaderamente Lucha Sabe que para Ganar hay que Perder Muchas Veces....”**

**Br. Shirley Maley Calderón Romero.**







## DEDICATORIA

Quiero dedicarle este trabajo **A Dios** que me ha dado la vida, la fortaleza, la fe, la salud y la esperanza para terminar este trabajo.

**A mis padres: Juan Francisco Iglesias Hernández y Doris de la Concepción Gutiérrez Ríos**, pilares fundamentales en mi vida, quienes a lo largo me han apoyado, motivado en mi formación académica, que creyeron en mí en todo momento y no dudaron de mis habilidades.

Sin ellos, jamás hubiese podido conseguir lo que hasta ahora he logrado. Su tenacidad y lucha insaciable han hecho de ellos el gran ejemplo a seguir y destacar, no solo para mí, sino para mis hermanos.

**A mi abuelita: Graciela Ríos**, por darme sus consejos y dedicación a lo largo de mi vida.

**A mis Hermanos: Axel Francisco Iglesias Gutiérrez y Yazer José Iglesias Gutiérrez**, por darme siempre su apoyo.

**A mis compañeras** de tesis porque en esta armonía grupal lo hemos logrado.

**A mi Tutor: Ing. Julio César González Moreno**, por su tiempo, esfuerzo y dedicación.

**Br. Yendri Yacenia Iglesias Gutiérrez.**





## ÍNDICE GENERAL.

<b><i>I. INTRODUCCIÓN.....</i></b>	<b><i>23</i></b>
<b><i>II. ANTECEDENTES.....</i></b>	<b><i>25</i></b>
<b><i>III. JUSTIFICACIÓN.....</i></b>	<b><i>27</i></b>
<b><i>IV. OBJETIVOS.....</i></b>	<b><i>29</i></b>
Objetivo General:.....	29
Objetivos Específicos:.....	29
<b><i>V. MARCO TEÓRICO.....</i></b>	<b><i>33</i></b>
Definiciones Básicas:.....	33
<b><i>VI. METODOLOGÍA.....</i></b>	<b><i>48</i></b>
Método a utilizar para el desarrollo de nuestro sistema.....	49
<b><i>VII. RECURSOS DISPONIBLES.....</i></b>	<b><i>52</i></b>
Hardware:.....	52
Software:.....	52
<b><i>VIII. ANÁLISIS DEL SISTEMA.....</i></b>	<b><i>54</i></b>
<b><i>IX. ESPECIFICACIÓN DE REQUISITOS SOFTWARE (ERS).....</i></b>	<b><i>56</i></b>
1-Introducción.....	56
2. Descripción General.....	59
3. Requisitos Específicos.....	61
<b><i>X. CASOS DE USO.....</i></b>	<b><i>79</i></b>
Roles de los Usuarios.....	81
<b><i>XI. DIAGRAMA DE CLASE.....</i></b>	<b><i>83</i></b>
<b><i>XII. DIAGRAMA ENTIDAD-RELACIÓN.....</i></b>	<b><i>85</i></b>
<b><i>XIII. RELACIÓN DE LAS TABLAS DE LA BBDD'S.....</i></b>	<b><i>87</i></b>
<b><i>XIV. DIAGRAMA DE FLUJOS DE DATOS.....</i></b>	<b><i>89</i></b>
Diagrama de Contexto.....	89
Nivel 1.....	89
Nivel 2.....	92





<b>XV. DICCIONARIO DE DATOS .....</b>	<b>95</b>
<b>XVI. DISEÑO DE DATOS .....</b>	<b>99</b>
<b>Tabla 1 Matricula.....</b>	<b>99</b>
<b>Tabla 2 Tabla Estudiante. ....</b>	<b>100</b>
<b>Tabla 3 Tabla Profesor.....</b>	<b>101</b>
<b>Tabla 4 Tabla Asignatura .....</b>	<b>102</b>
<b>Tabla 5 Tabla Plan de Estudio .....</b>	<b>102</b>
<b>Tabla 6 Tabla Notas .....</b>	<b>103</b>
<b>Tabla 7 Tabla para Contraseña de Usuarios.....</b>	<b>103</b>
<b>XVII. DISEÑO PROCEDIMENTAL.....</b>	<b>105</b>
<b>XVIII. DISEÑO ARQUITECTÓNICO .....</b>	<b>107</b>
<b>XIX. DISEÑO DE INTERFACES .....</b>	<b>109</b>
<b>Ventana de Presentación. ....</b>	<b>109</b>
<b>Ventana de inicio de Sesión.....</b>	<b>109</b>
<b>Ventana Principal .....</b>	<b>110</b>
<b>Registro de Estudiantes .....</b>	<b>110</b>
<b>Búsqueda de Estudiante para un Reingreso. ....</b>	<b>111</b>
<b>Registrar Profesor. ....</b>	<b>111</b>
<b>Ingresar Asignatura .....</b>	<b>112</b>
<b>Ingreso de Plan de Estudio. ....</b>	<b>112</b>
<b>Registrar Usuario.....</b>	<b>113</b>
<b>Realizar Copia de Seguridad. ....</b>	<b>113</b>
<b>Restaurar Copia de Seguridad. ....</b>	<b>114</b>
<b>Formulario Acerca de.....</b>	<b>114</b>
<b>Formulario de Notas.....</b>	<b>115</b>
<b>Formulario para creación de Boletines.....</b>	<b>115</b>
<b>Reporte de Notas. ....</b>	<b>116</b>
<b>Formulario para creación de Listado de Estudiantes. ....</b>	<b>116</b>





Reporte de Listado de Estudiante.....	117
Formulario para creación de una hoja de Matrícula.....	117
Reporte de Matrícula. ....	118
<b>XX. CONCLUSIÓN.....</b>	<b>120</b>
<b>XXI. RECOMENDACIONES.....</b>	<b>122</b>
<b>XXII. BIBLIOGRAFÍA.....</b>	<b>124</b>
Libros consultados:.....	124
Trabajos monográficos consultados: .....	124
Referencias de Internet:.....	124
<b>XXIII. ANEXOS .....</b>	<b>126</b>
Creación de la Base de Datos en Microsoft SQL Server Management Express.....	126
Códigos de ejemplo empelados en la aplicación .....	131
Encriptar una cadena de caracteres .....	131
Formulario MDI. ....	132
Expresión regular para quitar múltiples espacios en blanco.....	140
Reporte Listado de Estudiantes.....	140







## ÍNDICE DE FIGURAS

<b>Figura No.1 Arquitectura de la plataforma .NET Framework. ....</b>	<b>37</b>
<b>Figura No.2 Fases del Modelo en Cascada .....</b>	<b>49</b>
<b>Figura No.3 Especificación Conceptual del Programa .....</b>	<b>54</b>
<b>Figura No.3 Caso de uso Administrador.....</b>	<b>79</b>
<b>Figura No.4 Caso de Uso Usuario .....</b>	<b>80</b>
<b>Figura No.5 Roles de los Usuarios. ....</b>	<b>81</b>
<b>Figura No.6 Diagrama de Clase. ....</b>	<b>83</b>
<b>Figura No.7 Diagrama Entidad-Relación.....</b>	<b>85</b>
<b>Figura No.8 Relación entre Tablas. ....</b>	<b>87</b>
<b>Figura No.9 Diseño Procedimental.....</b>	<b>105</b>
<b>Figura No.10 Diseño Arquitectónico. ....</b>	<b>107</b>
<b>Figura No.11 Presentación.....</b>	<b>109</b>
<b>Figura No.12 Inicio de Sesión. ....</b>	<b>109</b>
<b>Figura No.13 Menú Principal. ....</b>	<b>110</b>
<b>Figura No.14 Registro de un Estudiante de Nuevo Ingreso. ....</b>	<b>110</b>
<b>Figura No.16 Registro de Profesor. ....</b>	<b>111</b>
<b>Figura No.17 Registro de Asignatura. ....</b>	<b>112</b>
<b>Figura No.18 Registro Plan de Estudio. ....</b>	<b>112</b>
<b>Figura No.19 Registro de Usuario. ....</b>	<b>113</b>
<b>Figura No.20 Realizar Copia de Seguridad. ....</b>	<b>113</b>
<b>Figura No.21 Restaurar Copia de Seguridad. ....</b>	<b>114</b>
<b>Figura No.22 Acerca de. ....</b>	<b>114</b>
<b>Figura No.24 Introducción de Datos para generar Certificados de Calificaciones.....</b>	<b>115</b>
<b>Figura No.25 Reporte de Calificaciones. ....</b>	<b>116</b>
<b>Figura No.26 Introducción de Datos para obtener Listado de Estudiantes. ....</b>	<b>116</b>
<b>Figura No.27 Reporte Listado de Estudiantes. ....</b>	<b>117</b>





<b>Figura No.28 Introducción de datos para generar Reporte de matrícula de un estudiante. ...</b>	<b>117</b>
<b>Figura No.29 Reporte de matrícula.....</b>	<b>118</b>
<b>Figura No.30 Microsoft SQL Server 2005.....</b>	<b>126</b>
<b>Figura No.31 Conectarse a Microsoft SQL Server 2005. ....</b>	<b>127</b>
<b>Figura No.32 Entorno de Microsoft SQL Server 2005. ....</b>	<b>127</b>
<b>Figura No.33 Crear una Base de Datos. ....</b>	<b>128</b>
<b>Figura No.34 Nombre a la Base de Datos. ....</b>	<b>128</b>
<b>Figura No.35 Agregar Campos a una tabla de la Base de Datos.....</b>	<b>129</b>
<b>Figura No.36 Poner Nombre a una tabla de la Base de Datos. ....</b>	<b>129</b>
<b>Figura No.37 Campos de una tabla de una Base de Datos.....</b>	<b>130</b>



**TEMA:**

**Automatización del Registro Académico del Colegio Apostólico  
Lirio de los Valles de la Ciudad de León.**





## I. INTRODUCCIÓN

En nuestros tiempos la educación constituye la base principal del desarrollo científico/cultural de todo ser humano, lo que permite que se incrementen cada vez más el número de personas que desean solventar esta necesidad tan básica.

Con el desarrollo de la tecnología se puede tener un mejor control de la información en instituciones, organizaciones, colegios, empresas y con cualquier usuario en general.

Con los avances tecnológicos se pueden crear sistemas que realicen un mejor control y gestión de la información de los estudiantes que pertenecen a los distintos centros educativos, por medio de un registro académico que les pueda facilitar el manejo de todas las notas y los datos de mayor relevancia de los estudiantes de forma automatizada. Por tanto en este trabajo monográfico se plantea el desarrollo de una aplicación para la Automatización del Registro Académico del Colegio Apostólico Lirio de los Valles de la Ciudad de León; dicha aplicación ha sido elaborada basándose en conocimientos teóricos/prácticos y habilidades adquiridas durante el transcurso de la carrera.







## II. ANTECEDENTES

Actualmente en los centros educativos se controla una gran cantidad de datos de estudiantes de reingreso y nuevo ingreso de cada año, esto provoca que la manipulación de toda la información tome mucho tiempo y pueda estar propenso a errores.

Con la creación de esta aplicación pretendemos alcanzar un estricto y riguroso control de la información de todos los estudiantes que se encuentran matriculados en el Colegio Apostólico Lirio de los Valles, facilitándole la tarea a la persona encargada de realizar esta labor.

Antes de la elaboración del presente trabajo monográfico el centro educativo Colegio Apostólico Lirio de los Valles no contaba con ninguna aplicación de automatización académica que realizará las tareas de una manera ordenada, ya que se trabajaba con una base de datos creada en Microsoft Access, pero el control de los estudiantes lo realizaban por medio de varios documento de Microsoft Word, uno por cada grado y de este documento obtenían el boletín de los alumnos.





### III. JUSTIFICACIÓN

Los procesos académicos llevados a cabo dentro del centro educativo Colegio Apostólico Lirio de los Valles han sido desarrollados de tal forma que la parte informática toma su papel únicamente como almacén de datos, más no así como una herramienta automatizada de gestión y control de dichos datos. Esto lleva a pensar que es necesaria y de suma importancia la elaboración de una aplicación de automatización y de gestión académica dentro del centro educativo. Dicha aplicación se encargará de facilitarle al personal académico un buen manejo de los datos de los estudiantes, datos de las notas, datos de los profesores, entre otros; todo ello a través de una gestión eficiente y confiable, que ofrece una respuesta rápida a diversas necesidades y que principalmente sea consistente y ordenada.

De esta manera se podrá almacenar adecuadamente toda la información académica, llevar a cabo un mejor control y manejo en las notas de los estudiantes, así como registrar un nuevo estudiante, y de esta manera poder realizar un certificado de notas para cada uno de los Estudiantes (Boletines) ya que será el único medio que utilizará el colegio para mostrar las notas (Boletines) y solucionar de esta manera la falta de un sistema informático de gestión académica que sea ordenado y consistente en este centro educativo.





## IV. OBJETIVOS

### *Objetivo General:*

- Elaborar e implementar una aplicación para la automatización del registro académico del centro educativo Colegio Apostólico Lirio de los Valles de la Ciudad de León.

### *Objetivos Específicos:*

- Diseñar la aplicación empleando conocimientos asociados a la Ingeniería del Software para obtener un producto software de calidad.
- Crear una interfaz de aplicación que sea atractiva e intuitiva para el usuario final.
- Ofrecer un acceso fácil y eficiente de los datos a través de la generación de reportes de cada uno de los elementos que son gestionados y administrados por el sistema.
- Brindar un menor tiempo de respuesta asociado a los procesos de matrícula y generación de notas llevados a cabo dentro del centro educativo.
- Permitir la realización de búsquedas de información académica, que son gestionadas por el sistema.
- Mostrar información básica asociada a los docentes que laboran en el centro educativo.
- Definir e implantar roles de usuario que permitan acceder y manipular de forma segura la información gestionada por el sistema.
- Asegurar la disponibilidad de la información gestionada por el sistema mediante la realización de copias de seguridad.





## V. MARCO TEÓRICO.







## V. MARCO TEÓRICO

### *Definiciones Básicas:*

#### 5.1 Sistema de Información

Conjunto de componentes que interaccionan entre sí para lograr un objetivo común.

Realiza cuatro actividades básicas:

1. **Entrada de información:** Proceso donde el sistema de información toma los datos de forma manual o automática que se requiere para procesar la información.
2. **Almacenamiento de información:** Proceso por el cual la información que se introduce suele ser almacenada en una estructura de información denominada Archivo.
3. **Procesamiento de información:** la capacidad del sistema de efectuar cálculos de acuerdo con las operaciones preestablecidas.
4. **Salida de información:** Capacidad de un sistema de información para sacar la información procesada o bien mostrar los datos al exterior.

#### 5.2 Programación Orientada a Objetos

##### 5.2.1 Conceptos de clase y objeto

Un **objeto** es un conjunto de atributos (almacenan los datos del objetos) y de métodos que permiten manipular dichos atributos.

Una **clase** es la definición de las características concretas de un determinado tipo de objeto. Es decir, de cuáles son los datos y los métodos de los que van a disponer todos los objetos de ese tipo. Se suele decir que el **tipo de dato** de un objeto es la clase que define las características del mismo.



### 5.2.2 Definición de Programación Orientada a Objetos.

Programación orientada a objetos (POO) es un modelo de programación que utiliza Objetos, ligados mediante mensajes, para la solución de problemas.

### 5.2.3 Pilares básicos de la programación orientada a objetos.

- **La Encapsulación:** Los datos se ocultan dentro de una clase y la clase implementa un diseño que permite que otras partes del código accedan a esos datos de forma eficiente. El nombre del método reside en la clase base y los distintos cuerpos del método residen en las clases derivadas
- **La Herencia:** Mecanismo que permite definir nuevas clases a partir de otras ya definidas, de modo que si en la definición de una clase indicamos que ésta hereda de otra, entonces la primera **-clase hija-** será tratada por el compilador automáticamente como si su definición incluyese la definición de la segunda - **clase padre** o **clase base**. Mediante el mecanismo de herencia se contribuye a la reutilización del código, algo habitual en todos los lenguajes orientados a objetos. Donde la herencia es una relación entre clases y las nuevas clases añaden especialización a las existentes.

La herencia se divide en dos:

*1-Herencia Simple:* que permite que una clase se derive de una sola clase base.

*2-Herencia Múltiple:* permite que una clase se derive de más de una clase base.

- **El Polimorfismo:** Capacidad de almacenar objetos de un determinado tipo en variables de tipos antecesores del primero a costa, claro está, de sólo poderse acceder a través de dicha variable a los miembros comunes a ambos tipos.

### 5.2.4 Importancia de la Programación Orientada a Objeto.



La importancia de trabajar con programación orientada a objetos es no expresar las ideas en términos de bloques de código que actúan sobre un grupo de variables, en su lugar expresan las ideas en términos de objetos, clases, mensajes y métodos que pueden realizar distintas tareas programadas, esta misma posee mecanismos y características que hacen que la creación y el uso de objetos sean sencillos y flexible.

### 5.2.5 Ventaja de la POO sobre la Programación Estructurada

- Se puede especificar lo que debe hacer el software.
- Se usan objetos que pertenecen a clases.
- Durante la implementación podemos tener ámbitos de nivel de clases que permite ocultar variables en las definiciones de clases.

## 5.3 Plataforma .NET

### 5.3.1 ¿Qué es Microsoft .NET?

Microsoft.NET es el conjunto de nuevas tecnologías en las que Microsoft ha estado trabajando durante los últimos años con el objetivo de obtener una plataforma sencilla y potente para distribuir el software en forma de servicios que puedan ser suministrados remotamente y que puedan comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados. Ésta es la llamada **plataforma .NET**, y a los servicios antes comentados se les denominan **servicios Web**.

Para crear aplicaciones para la plataforma .NET, tanto servicios Web como aplicaciones tradicionales (aplicaciones de consola, aplicaciones de ventanas, servicios de Windows NT, etc.), Microsoft ha publicado el denominado kit de desarrollo de software conocido como **.NET Framework SDK**, que incluye las herramientas necesarias tanto para su desarrollo como para su distribución y ejecución y **Visual Studio .NET**.



### 5.3.2 Arquitectura de la plataforma .NET Framework

Los programas de C Sharp se ejecutan sobre el .NET Framework, un componente que forma parte de Windows y que incluye un sistema de ejecución virtual denominado Common Language Runtime (CLR) y un conjunto unificado de bibliotecas de clases. CLR es la implementación comercial de Microsoft de Common Language Infrastructure (CLI), norma internacional que constituye la base para crear entornos de ejecución y desarrollo en los que los lenguajes y las bibliotecas trabajan juntos sin problemas.

El código fuente escrito en C Sharp se compila en un lenguaje intermedio (IL) conforme con la especificación CLI. El código de lenguaje intermedio, junto con recursos tales como mapas de bits y cadenas, se almacena en disco en un archivo ejecutable denominado ensamblado, cuya extensión es .exe o .dll. Un ensamblado contiene un manifiesto que ofrece información sobre los tipos, la versión, la referencia cultural y los requisitos de seguridad del ensamblado.

Cuando se ejecuta un programa de C Sharp, el ensamblado se carga en CLR, con lo que se pueden realizar diversas acciones en función de la información del manifiesto. A continuación, si se cumplen los requisitos de seguridad, CLR realiza una compilación Just In Time (JIT) para convertir el código de lenguaje intermedio en instrucciones máquina nativa. CLR también proporciona otros servicios relacionados con la recolección automática de elementos no utilizados, el control de excepciones y la administración de recursos. El código ejecutado por CLR se denomina algunas veces "código administrado", en contraposición al "código no administrado" que se compila en lenguaje máquina nativo destinado a un sistema específico. En el diagrama siguiente se muestran las relaciones en tiempo de compilación y tiempo de ejecución de los archivos de código fuente de C Sharp, las bibliotecas de clases base, los ensamblados y CLR.

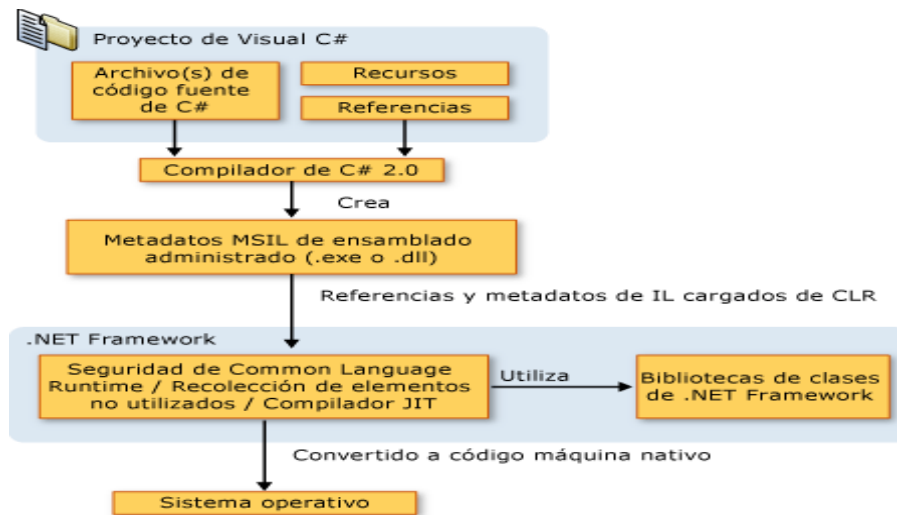


Figura No.1 Arquitectura de la plataforma .NET Framework.

La interoperabilidad del lenguaje es una función clave de .NET Framework. Como el código de lenguaje intermedio generado por el compilador de C Sharp cumple la especificación de tipos común (CTS), este código generado en C Sharp puede interactuar con el código generado en las versiones .NET de Visual Basic, Visual C++, Visual J# o cualquiera de los más de 20 lenguajes conformes con CTS. Un único ensamblado puede contener varios módulos escritos en diferentes lenguajes .NET, y los tipos admiten referencias entre sí como si estuvieran escritos en el mismo lenguaje.

Además de los servicios en tiempo de ejecución, .NET Framework también incluye una amplia biblioteca de más de 4.000 clases organizada en espacios de nombres que ofrecen una diversidad de funciones útiles para la entrada y salida de archivos, la manipulación de cadenas, el análisis de archivos XML y los controles de formularios Windows Forms. La aplicación de C Sharp típica utiliza continuamente la biblioteca de clases de .NET Framework para el tratamiento de las tareas comunes de "infraestructura".

#### 5.4. Visual Studio.

Fue utilizado para la creación de interfaces del software, el cual es un conjunto completo de herramientas de desarrollo integrado para construir aplicaciones en las que el diseño de su interfaz grafica se puede realizar de una manera visual de tal manera que se puedan solo arrastrar los



controles que se necesiten sobre la ventana destino de los mismos y en la que se pueden utilizar varios lenguajes de programación tales como Visual C++, Visual C Sharp, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y servicios web, sobre cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.

### **5.5 Crystal Reports**

Visual Studio .NET incluye Crystal Reports como un mecanismo "oficial" de creación de reportes, el cual es un producto muy desarrollado, complejo, y cuenta con una infinidad de opciones.

Herramienta utilizada para la creación de reportes en la aplicación.

### **5.6 Lenguaje C Sharp**

C Sharp es un lenguaje orientado a objetos elegante y con seguridad de tipos que permite a los desarrolladores crear una amplia gama de aplicaciones sólidas y seguras que se ejecutan sobre .NET Framework. Es posible utilizar este lenguaje para crear aplicaciones cliente para Windows tradicionales, servicios Web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones con acceso a bases de datos, y muchas otras más.

### **5.7 Base de Datos**

#### **5.7.1 ¿Qué es una Base de Datos?**

Una base de datos es un "almacén" que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar, utilizar y modificar fácilmente.

#### **5.7.2. Definición de base de datos**



Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

### **5.7.3. Características**

Entre las principales características de los sistemas de base de datos podemos mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

### **5.7.4. Importancia de los Sistemas Gestores de Base de Datos (SGBD)**

Los Sistemas Gestores de Base de Datos (en inglés DataBase Management System) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

### **5.7.5. Ventajas de las bases de datos**

- **Control sobre la redundancia de datos:**

Los sistemas de ficheros almacenan varias copias de los mismos datos en ficheros distintos. Esto hace que se desperdicie espacio de almacenamiento, además de provocar la falta de consistencia de datos.



En los sistemas de bases de datos todos estos ficheros están integrados, por lo que no se almacenan varias copias de los mismos datos. Sin embargo, en una base de datos no se puede eliminar la redundancia completamente, ya que en ocasiones es necesaria para modelar las relaciones entre los datos.

- **Consistencia de datos:**

Eliminando o controlando las redundancias de datos se reduce en gran medida el riesgo de que haya inconsistencias. Si un dato está almacenado una sola vez, cualquier actualización se debe realizar sólo una vez, y está disponible para todos los usuarios inmediatamente. Si un dato está duplicado y el sistema conoce esta redundancia, el propio sistema puede encargarse de garantizar que todas las copias se mantengan consistentes.

- **Compartición de datos:**

En los sistemas de ficheros, los ficheros pertenecen a las personas o a los departamentos que los utilizan. Pero en los sistemas de bases de datos, la base de datos pertenece a la empresa y puede ser compartida por todos los usuarios que estén autorizados.

- **Mantenimiento de estándares:**

Gracias a la integración es más fácil respetar los estándares necesarios, tanto los establecidos a nivel de la empresa como los nacionales e internacionales. Estos estándares pueden establecerse sobre el formato de los datos para facilitar su intercambio, pueden ser estándares de documentación, procedimientos de actualización y también reglas de acceso.

- **Mejora en la integridad de datos:**

La integridad de la base de datos se refiere a la validez y la consistencia de los datos almacenados. Normalmente, la integridad se expresa mediante restricciones o reglas que no se pueden violar. Estas restricciones se pueden aplicar tanto a los datos, como a sus relaciones, y es el SGBD quien se debe encargar de mantenerlas.





- **Mejora en la seguridad:**

La seguridad de la base de datos es la protección de la base de datos frente a usuarios no autorizados. Sin unas buenas medidas de seguridad, la integración de datos en los sistemas de bases de datos hace que éstos sean más vulnerables que en los sistemas de ficheros.

- **Mejora en la accesibilidad a los datos:**

Muchos SGBD proporcionan lenguajes de consultas o generadores de informes que permiten al usuario hacer cualquier tipo de consulta sobre los datos, sin que sea necesario que un programador escriba una aplicación que realice tal tarea.

- **Mejora en la productividad:**

El SGBD proporciona muchas de las funciones estándar que el programador necesita escribir en un sistema de ficheros. A nivel básico, el SGBD proporciona todas las rutinas de manejo de ficheros típicas de los programas de aplicación.

El hecho de disponer de estas funciones permite al programador centrarse mejor en la función específica requerida por los usuarios, sin tener que preocuparse de los detalles de implementación de bajo nivel.

- **Mejora en el mantenimiento:**

En los sistemas de ficheros, las descripciones de los datos se encuentran inmersas en los programas de aplicación que los manejan.

Esto hace que los programas sean dependientes de los datos, de modo que un cambio en su estructura, o un cambio en el modo en que se almacena en disco, requiere cambios importantes en los programas cuyos datos se ven afectados.



Sin embargo, los SGBD separan las descripciones de los datos de las aplicaciones. Esto es lo que se conoce como independencia de datos, gracias a la cual se simplifica el mantenimiento de las aplicaciones que acceden a la base de datos.

- **Aumento de la concurrencia:**

En algunos sistemas de ficheros, si hay varios usuarios que pueden acceder simultáneamente a un mismo fichero, es posible que el acceso interfiera entre ellos de modo que se pierda información o se pierda la integridad. La mayoría de los SGBD gestionan el acceso concurrente a la base de datos y garantizan que no ocurran problemas de este tipo.

- **Mejora en los servicios de copias de seguridad:**

Muchos sistemas de ficheros dejan que sea el usuario quien proporcione las medidas necesarias para proteger los datos ante fallos en el sistema o en las aplicaciones. Los usuarios tienen que hacer copias de seguridad cada día, y si se produce algún fallo, utilizar estas copias para restaurarlos.

En este caso, todo el trabajo realizado sobre los datos desde que se hizo la última copia de seguridad se pierde y se tiene que volver a realizar. Sin embargo, los SGBD actuales funcionan de modo que se minimiza la cantidad de trabajo perdido cuando se produce un fallo.

#### 5.7.6. Desventajas de las bases de datos

- **Complejidad:**

Los SGBD son conjuntos de programas que pueden llegar a ser complejos con una gran funcionalidad. Es preciso comprender muy bien esta funcionalidad para poder realizar un buen uso de ellos.



- **Coste del equipamiento adicional:**

Tanto el SGBD, como la propia base de datos, pueden hacer que sea necesario adquirir más espacio de almacenamiento. Además, para alcanzar las prestaciones deseadas, es posible que sea necesario adquirir una máquina más grande o una máquina que se dedique solamente al SGBD. Todo esto hará que la implantación de un sistema de bases de datos sea más cara.

- **Vulnerable a los fallos:**

El hecho de que todo esté centralizado en el SGBD hace que el sistema sea más vulnerable ante los fallos que puedan producirse. Es por ello que deben tenerse copias de seguridad (Backup).

### 5.7.7 Tipos de Campos de las Bases de Datos

Cada Sistema de Base de Datos posee tipos de campos que pueden ser similares o diferentes. Entre los más comunes podemos nombrar:

- **Numérico:** entre los diferentes tipos de campos numéricos podemos encontrar enteros “sin decimales” y reales “decimales”.
- **Booleanos:** poseen dos estados: Verdadero “Si” y Falso “No”.
- **Memos:** son campos alfanuméricos de longitud ilimitada. Presentan el inconveniente de no poder ser indexados.
- **Fechas:** almacenan fechas facilitando posteriormente su explotación. Almacenar fechas de esta forma posibilita ordenar los registros por fechas o calcular los días entre una fecha y otra.
- **Alfanuméricos:** contienen cifras y letras. Presentan una longitud limitada (255 caracteres).
- **Auto incrementables:** son campos numéricos enteros que incrementan en una unidad su valor para cada registro incorporado. Su utilidad resulta: Servir de identificador ya que resultan exclusivos de un registro.



### 5.7.8. Tipos de Base de Datos

Entre los diferentes tipos de base de datos, podemos encontrar los siguientes:

- **MySql:** es una base de datos con licencia GPL basada en un servidor. Se caracteriza por su rapidez. No es recomendable usar para grandes volúmenes de datos.
- **PostgreSQL y Oracle:** Son sistemas de base de datos poderosos. Administra muy bien grandes cantidades de datos, y suelen ser utilizadas en intranets y sistemas de gran calibre.
- **Access:** Es una base de datos desarrollada por Microsoft. Esta base de datos, debe ser creada bajo el programa access, el cual crea un archivo .mdb con la estructura ya explicada.
- **Microsoft SQL Server 2005:** Es una base de datos más potente que access desarrollada por Microsoft. Se utiliza para manejar grandes volúmenes de informaciones. Esta es la base de datos utilizada para la solución de la aplicación.
- **5.7.9. Estructura de una Base de Datos**

Una base de datos, a fin de ordenar la información de manera lógica, posee un orden que debe ser cumplido para acceder a la información de manera coherente. Cada base de datos contiene una o más tablas, que cumplen la función de contener los campos.

El lenguaje SQL es el más universal en los sistemas de base de datos. Este lenguaje nos permite realizar consultas a nuestras bases de datos para mostrar, insertar, actualizar y borrar datos.

A continuación veremos un ejemplo de ellos:

<b>Mostrar:</b> para mostrar los registros se utiliza la instrucción	Select. Select * From comentarios
<b>Insertar:</b> los registros pueden ser introducidos a partir de sentencias que emplean la instrucción Insert.	Insert Into comentarios (titulo, texto, fecha) Values ('saludos', 'como esta', '22-10-2007')
<b>Borrar:</b> Para borrar un registro se utiliza la instrucción Delete. En este caso se especifica cuál o cuáles son los registros que queremos borrar. Se debe realizar una selección con Where.	Delete From comentarios Where id='1'
<b>Actualizar:</b> para actualizar los registros se	Update comentarios Set titulo='Mi Primer



utiliza la instrucción Update y se tiene que especificar cuáles son los nuevos valores de los campos que deseamos actualizar.	Comentario' Where id='1'.
---	---------------------------

## 5.8 Características de SQL Server Express.

### Compatibilidad de las características de SQL Server Express

- SQL Server Express admite Service Broker, pero no admite la comunicación directa entre dos servidores SQL Server Express.
- SQL Server Express es compatible con la mayoría de las características y funciones de SQL Server.

En la tabla siguiente se muestran algunas de las principales características y componentes que admite.

Procedimientos almacenados	Administrador de configuración de SQL Server
Vistas	Replicación (sólo como suscriptor)
Desencadenadores	Optimizador de consultas avanzado
Cursores	SMO/RMO
Utilidades <code>sqlcmd</code> y <code>sql</code> .	Integración con Visual Studio 2005
Niveles de aislamiento de instantánea	Service Broker (únicamente como cliente) <sup>1</sup>
Compatibilidad con XML nativo, incluidos XQuery y esquemas XML	CLR de SQL
Compatibilidad con lenguajes de Transact-SQL	Conjuntos de resultados activos múltiples (MARS)
Conexión de administrador dedicada <sup>2</sup>	Asistente para importación y exportación

## 6. Help Maker

Es una aplicación para crear archivos de ayuda sin tener que manejar archivos múltiples. HelpMaker hace a los archivos WinHelp, DE RTF y DE HTML-HELP. Programa utilizado para crear el manual de usuario en la aplicación.



## **7. Creación del Documento.**

### **7.1 Easy Case.**

Es una aplicación CASE que permite realizar el modelado de diagramas de diferente tipo; entre los más importantes tenemos: Diagrama E-R, diagrama de flujo de datos, entre otros. Este programa fue utilizado para la creación de los diagramas de Flujo de Datos.

### **7.2 Dia**

Es una aplicación para crear diagramas técnicos. Su interfaz y características son holgadamente estructuradas con pautas repetidas después del Programa de Windows Visio. Las características de Dia incluyen página múltiple escribiendo en letras de imprenta, exportación para muchos formatos (los discos de corta duración, SVG, meta archivero informático de gráficos y PNG), y la habilidad para usar formas de costumbre creadas por el usuario como descripciones simples XML. Dia es útil para dibujar diagramas UML, mapas de la red, y diagramas de flujo.

Este programa fue utilizado para la creación del Diagrama Entidad Relación (E-R).

### **7.3 ZScreen**

Es una aplicación que facilita la captura de pantallas del escritorio o de las ventanas que tengamos activas en el ordenador. Ofrece una interfaz sencilla y fácil de utilizar. Este programa fue utilizado para realizar las capturas de pantalla de la aplicación y poder de esta manera colocar dichas capturas de pantalla en el manual de usuario.



## VI. METODOLOGÍA



## VI. METODOLOGÍA

Para poder elaborar este proyecto nos hemos basado en la información que nos brinda el centro educativo Colegio Lirio de los Valles obtenidos mediante entrevistas al director y la secretaria de dicho centro y por medio de la información que nos proporciona la Ingeniería del Software, la que nos facilita la elaboración del sistema en cada uno de los procesos y el desarrollo de su ciclo de vida, así como cada uno de los procedimientos y etapas en los que se dividirá la creación, el desarrollo, implementación, funcionamiento y mantenimiento del sistema.

Los elementos claves de la Ingeniería del Software son tres: **Métodos**, **Herramientas** y **Procedimientos**.

Por medio de la utilización de métodos se nos indicará como construir el software con la ayuda de diferentes modelos (estructurado, cascada, espiral, etc.) los que nos proporcionarán una serie de pasos que se deben seguir durante el proceso de desarrollo.

Para la elaboración de nuestro sistema implementaremos uno de los tres métodos que existen para el diseño de software como son:

- Método del ciclo de vida clásico.
- Método de desarrollo del análisis estructurado.
- Método del prototipo de sistemas.





### *Método a utilizar para el desarrollo de nuestro sistema.*

Para el desarrollo de nuestro sistema utilizaremos el modelo de ciclo de vida clásico en cascada que nos permite un enfoque sistemático donde se descomponen los procesos del desarrollo del sistema en diferentes fases donde la salida de un proceso es la entrada requerida para el siguiente proceso, en cada etapa del sistema como son: análisis, diseño, codificación, prueba y mantenimiento.

#### **CICLO DE VIDA DEL MODELO EN CASCADA**

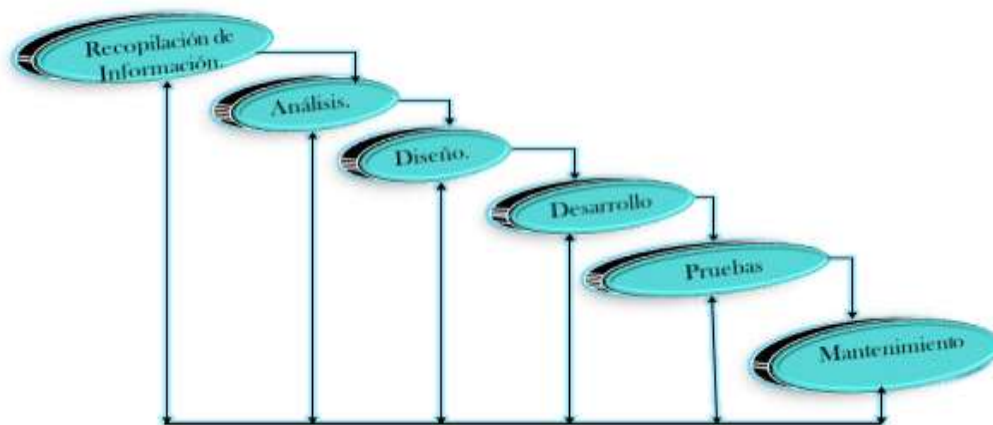


Figura No.2 Fases del Modelo en Cascada

#### **Actividades del Ciclo de Vida en Cascada.**

**Recopilación de información:** se realiza el estudio y recopilación de datos para obtener una visión de lo que será el sistema y que se desarrollará en él y satisfacer las necesidades de los usuarios finales.

**Análisis:** se realiza el análisis por medio de la especificación de requisito software (ERS) y con la ayuda de herramientas como:

- El diagrama de flujos de datos (DFD) que representa el flujo de la información desde la entrada hasta la salida de los datos.
- El diccionario de datos (DD) que almacenará los detalles y descripciones del flujo de dato y procesos que se llevaran a cabo.



**Diseño:** se realiza la definición de arquitecturas, interfaces MDI, estructuras de datos y la transformación del diagrama entidad relación (ER) en un diseño de base de datos.

**Desarrollo:** se realiza la transformación del diseño a un lenguaje de programación.

**Prueba:** se realiza la prueba para ver el correcto funcionamiento del sistema y poder identificar los errores que se pudieron presentar en etapa de la elaboración del sistema, para que se puedan cumplir con los requerimientos de los usuarios finales.

**Mantenimiento:** se realiza para producir algunos cambios en el sistema como son los de corrección, adaptación y mejoras. En esta fase se realizan los pasos de la fase de definición y de desarrollo, pero en contexto de un software que ya existe.



## VII. RECURSOS DISPONIBLES



## VII. RECURSOS DISPONIBLES

Para la creación y elaboración del software utilizaremos recursos disponibles como son las siguientes herramientas:

### *Hardware:*

Dos computadoras con las siguientes características:

- Procesador: Intel Pentium Dual-Core
- Memoria RAM: 2GB
- Disco Duro: 160GB
- Monitor: SONY

### *Software:*

- Microsoft Windows 7 Ultimate
- Microsoft Office 2007
- Easy Case
- Rational Rose
- Microsoft Visual Studio 2005 (Visual C sharp.Net)
- Microsoft SQL Server 2005
- SQL Server 2005 Management Studio Express
- Help Maker 7.4.4.0
- Día 0.97.1.
- Crystal Report
- ZScreen 3.28.3.1 Beta



## VIII. FASE DE ANÁLISIS



## VIII. ANÁLISIS DEL SISTEMA

En el siguiente diagrama se muestra de forma gráfica la especificación conceptual de nuestro sistema.

### ESPECIFICACIÓN CONCEPTUAL DEL PROGRAMA

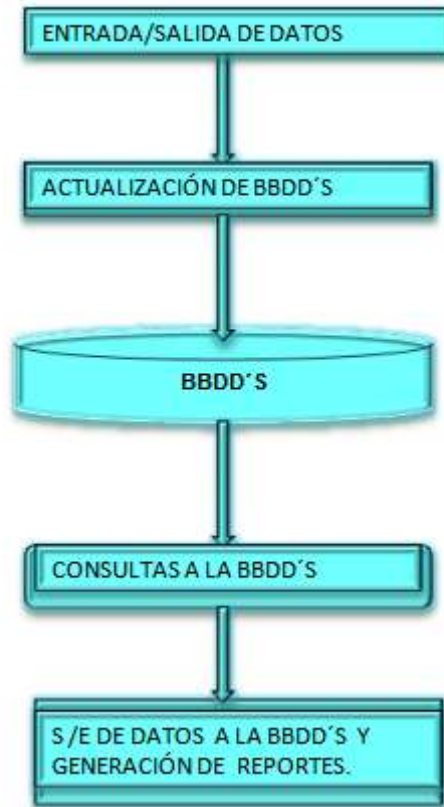


Figura No.3 Especificación Conceptual del Programa



# IX. ESPECIFICACIÓN DE REQUISITOS SOFTWARE



## IX. ESPECIFICACIÓN DE REQUISITOS SOFTWARE (ERS)

### *1-Introducción*

#### **1.1-Propósito.**

Definir el conjunto de especificaciones de requisitos software que debe cumplir la aplicación para automatizar los procesos del área de registro académico del Colegio Apostólico Lirio de los Valles.

El documento se encuentra dirigido al director del centro educativo y así como a los usuarios finales que tengan consentimiento del director del centro para utilizar el sistema. Este documento debe ser estudiado por los usuarios para un correcto y eficiente uso de la aplicación.

#### **1.2 -Alcance.**

El nombre con el que se conocerá a la aplicación será: “Automatización del Registro Académico del Colegio Apostólico Lirio de los Valles de la Ciudad de León”.

El producto realizará las siguientes funciones:

1. Administrar cuentas de usuario.
2. Gestionar cuentas de usuario.
3. Captura de los datos para el ingreso de estudiantes.
  - Nuevo Ingreso.
  - Reingreso.
4. Captura de los datos para el ingreso de profesores.
5. Captura de los datos para el ingreso de asignaturas.
6. Capturas de los datos para el ingreso de un plan de estudio.
7. Agregar las calificaciones de los estudiantes.





8. Generación de reportes de información.

- Reporte de lista de estudiantes.
- Reporte de matricula por estudiante.
- Reporte de calificaciones por estudiante.

9. Respaldo de la Base de Datos.

10. Restaurar Base de Datos.

**1.3 Definiciones, acrónimos y abreviaturas.**

- **Estudiante:** Ente que proporciona los datos de un estudiante. Se conocerá como: **estudiante.**
- **Profesor:** Persona que brinda sus servicios al centro educativo. Se conocerá como: **profesor.**
- **Asignatura:** Entidad que mantendrá la información de los datos de las asignaturas que serán impartidas. Se conocerá como: **nombre\_asignatura.**
- **Calificaciones:** Entidad que mantendrá información de las notas de los estudiantes. Se conocerá como: **calificaciones.**
- **Usuario:** Persona natural que a través de su cuenta de usuario accede al sistema de información. Se conocerá como: **usuario**
- **Cuenta de Usuario:** Entidad que mantendrá información de los usuarios que puedan tener acceso al sistema. Se conocerá como: **cuenta de usuario.**
- **Clave de Acceso:** Entidad con la que el usuario podrá acceder al sistema. Se conocerá como: **contraseña.**
- **Roles:** Entidad que especifica los privilegios entre los usuarios que tengan acceso al sistema. Se conocerá como: **rol.**



- **Plan de Estudio:** Entidad que nos proporcionará saber a qué plan de estudio pertenece una asignatura. Se le conocerá como: **plan de estudio**.
- **Primer Bimestre:** es un dato que almacenará el valor de la nota del primer parcial de un estudiante. Se conocerá como: **IB**.
- **Segundo Bimestre:** es un dato que almacenará el valor de la nota del segundo parcial de un estudiante. Se conocerá como: **IIB**.
- **Tercer Bimestre:** es un dato que almacenará el valor de la nota del tercer parcial de un estudiante. Se conocerá como: **IIIB**.
- **Cuarto Bimestre:** es un dato que almacenará el valor de la nota del segundo parcial de un estudiante. Se conocerá como: **IVB**.
- **Primer Semestre:** es un dato calculado que almacenará el promedio de la nota del primer semestre. Se conocerá como: **[IS]**.
- **Segundo Semestre:** es un dato calculado que almacenará el promedio de la nota del segundo semestre. Se conocerá como: **IIS**.
- **Nota Final:** es un dato calculado que almacenará el promedio de la nota del primer semestre y del segundo semestre. Se conocerá como: **NF**.

#### 1.4 Referencias.

- Folleto de Análisis y Diseño de Sistema.
- Folleto de Especificación de Requisitos Software.
- Entrevista con el director y la secretaria del centro educativo Colegio Apostólico Lirio de los Valles.



## 1.5 Visión General.

Primeramente se realizará una descripción general del producto que se desea desarrollar para pasar posteriormente a estudiar cada uno de los Requisitos Específicos Individualmente.

## 2. Descripción General.

### 2.1 Relaciones del producto.

La aplicación se desarrollará e implementará en un equipo con las siguientes características:

- Procesador: Intel Pentium Dual-Core
- Memoria RAM: 2GB
- Disco Duro: 160GB
- Monitor: SONY

### 2.2 Funciones del producto

El producto software debe realizar las siguientes funciones:

- Cuando un usuario desee acceder al sistema deberá introducir su contraseña la cual el sistema validará, si después de tres intentos la contraseña no coincide no podrá acceder al sistema.
- Se podrán modificar las cuentas de todos los usuarios excluyendo la cuenta del usuario administrador.
- Cuando se hace el ingreso de un estudiante, si es un nuevo ingreso se deberán ingresar los datos necesarios para poder registrarlo y si es un reingreso deberá realizar una búsqueda por apellido del estudiante donde al final se generará un reporte de dicha matrícula.
- Cuando se desee agregar un profesor, se debe introducir los datos necesarios.
- Cuando se desee agregar una asignatura, se debe introducir los datos necesarios.
- Cuando se desee agregar un plan de estudio, se debe introducir los datos necesarios.



- Cuando se desee registrar las calificaciones de un estudiante, se debe introducir los datos necesarios.
- El usuario podrá disponer de los siguientes reportes:
  - Reporte de lista de estudiantes.
  - Reporte de matricula por estudiante.
  - Reporte de calificaciones por estudiante.
- Cuando se desee realizar un respaldo de la base de datos el usuario deberá introducir los datos necesarios.
- Cuando se desee restaurar la base de datos el usuario deberá introducir los datos necesarios.

### **2.3 Característica de los Usuarios.**

El usuario final de la aplicación puede tener conocimientos informáticos mínimos, motivo por el cual se debe incluir ayuda en el producto final.

### **2.4 Restricciones Generales.**

- El lenguaje de programación utilizado será C Sharp el cual utiliza un enfoque de programación orientada a objetos.
- El sistema gestor de la Base de Datos es SQL Server Management Studio Express.
- Microsoft Visual Studio 2005.
- Sistema Operativo Windows 7 Ultimate.
- Crystal Report.
- Rational Rose 98 Evaluation Edition.
- ZScreen 3.28.3.1 Beta
- HelpMaker 7.4.4.0 (Para la creación de ayuda).



- Día 0.97.1.

### *3. Requisitos Específicos.*

#### **3.1 Requisitos Funcionales.**

##### **3.1.1 Registrar cuenta de usuario a la aplicación.**

###### **3.1.1.1 Especificación**

###### **3.1.1.1.1 Introducción**

Esta función permitirá agregar una cuenta de usuario y la guardará en la base de datos.

###### **3.1.1.1.2 Entradas**

###### **Por Teclado:**

- Usuario.
- Contraseña.

###### **Por el Sistema: Rol.**

###### **3.1.1.1.3 Proceso**

Se mostrara por pantalla el formulario para introducir los datos de la cuenta de usuario. El rol podrá ser seleccionado de una lista.

###### **Los datos necesarios a introducir son:**

- usuario: Es un dato obligatorio. No debe existir en la base de datos.
- Contraseña: Es un dato obligatorio.

###### **3.1.1.1.4 Salida.**

Todos los datos introducidos se guardarán en la base de datos.



### **3.1.2 Modificar cuenta de usuario.**

#### **3.1.2.1 Especificación**

##### **3.1.2.1.1 Introducción**

Esta función permitirá modificar contraseña y rol de una cuenta de usuario y la actualizará en la base de datos.

##### **3.1.2.1.2 Entradas**

**Por teclado:**

- Contraseña.
- Confirmar Contraseña.

**Por el Sistema:**

- Nombre de Usuario.
- Rol.

**Los datos necesarios a introducir son:**

- Contraseña: Es un dato obligatorio.
- Confirmar Contraseña: Es un dato obligatorio.

##### **3.1.2.1.3 Proceso**

Se mostrará en la pantalla un formulario donde primeramente se tendrá que seleccionar el nombre de usuario a modificar, se introducirá la nueva contraseña si así se desea o se seleccionará el nuevo rol.

##### **3.1.2.1.4 Salida.**

Todos los datos introducidos se actualizarán en la base de datos.



### **3.1.3 Eliminar una cuenta de usuario de la aplicación.**

#### **3.1.3.1 Especificación**

##### **3.1.3.1.1 Introducción**

Esta función permitirá eliminar una cuenta de usuario y la borrará de la base de datos.

##### **3.1.3.1.2 Entradas**

**Por teclado:**

- Nombre de Usuario.

**Los datos necesarios a introducir son:**

- Nombre de Usuario. Es un dato obligatorio.

##### **3.1.3.1.3 Proceso**

Se mostrará en la pantalla un formulario donde se tendrá que introducir el nombre de usuario a eliminar.

##### **3.1.3.1.4 Salida.**

Se eliminará la cuenta de usuario de la base de datos.

### **3.1.4 Validar entrada de una cuenta de usuario a la aplicación.**

#### **3.1.4.1 Especificación**

##### **3.1.4.1.1 Introducción**

Este proceso realizará la captura de todos los datos de una cuenta de usuario y verificará en la base de datos automáticamente.



### **3.1.4.1.2 Entradas**

#### **Por teclado:**

- Nombre de Usuario.
- Contraseña.

#### **Los datos necesarios a introducir son:**

- Nombre de Usuario: Es un dato obligatorio. Debe existir en la base de datos.
- Contraseña: Es un dato obligatorio. Debe existir en la base de datos.

### **3.1.4.1.3 Proceso**

Al iniciar se mostrará un formulario que permitirá introducir el nombre de usuario y su respectiva contraseña. El sistema verificará el nombre de usuario y contraseña, si estos están correctos, tendrá acceso a todas las funciones de la aplicación, si esta cuenta esta registrada con el rol de Administrador, si está registrada con el rol de Usuario tendrá acceso a ciertas funciones de la aplicación.

Si los datos introducidos son incorrectos podrá introducirlos nuevamente, si después de tres intentos los datos son erróneos la aplicación se cerrará.

### **3.1.4.1.4 Salida.**

Podrá acceder a la aplicación.

## **3.1.5 Captura de los datos para el nuevo ingreso de un estudiante.**

### **3.1.5.1 Especificación**

#### **3.1.5.1.1 Introducción**

Se realiza para agregar un estudiante en la base de datos del sistema.





### 3.1.5.1.2 Entradas

#### Por teclado:

- Nombres.
- Apellidos.
- Teléfono.
- Municipio.
- Año de Nacimiento.
- Dirección de Residencia.
- Observaciones.
- Código de Matricula.

#### Por el Sistema:

- Número de Carnet.
- Sexo.
- Departamento de Nacimiento.
- Día de Nacimiento.
- Mes de Nacimiento.
- Edad.
- Grado que Cursa.
- Modalidad.
- Situación Académica.



- Plan de Estudio.
- Año Lectivo.
- Fecha de Matricula.
- Asignaturas a llevar por el estudiante.

**Los datos necesarios a introducir son:**

- Nombres: Es un dato obligatorio.
- Apellidos: Es un dato obligatorio.
- Municipio: Es un dato obligatorio.
- Año de Nacimiento: Es un dato obligatorio.
- Dirección de Residencia: Es un dato obligatorio.
- Código de Matricula: Es un dato obligatorio.

**3.1.5.1.3 Proceso**

Se mostrará la pantalla de introducción de datos al usuario. El Número de Carnet, Edad, Situación Académica, Plan de Estudio, Año Lectivo, Fecha de Matricula y Asignaturas a llevar por el estudiante aparecerán automáticamente. El Sexo, Departamento de Nacimiento, Día de Nacimiento, Mes de Nacimiento, Grado que Cursa y Modalidad podrán ser seleccionados de una lista. El resto de datos deberán ser introducidos por teclado.

**3.1.5.1.4 Salida.**

Se generará un reporte de matrícula del estudiante.

**3.1.6 Captura de los datos para el reingreso de un estudiante.**

**3.1.6.1 Especificación**

**3.1.6.1.1 Introducción**

Se realiza para registrar una matrícula de reingreso.



### 3.1.6.1.2 Entradas

#### Por teclado:

- Código de Matrícula.
- Observaciones.

#### Por el Sistema:

- Número de Carnet.
- Nombres.
- Apellidos.
- Teléfono.
- Municipio.
- Año de Nacimiento.
- Dirección de Residencia.
- Observaciones.
- Sexo.
- Departamento de Nacimiento.
- Día de Nacimiento.
- Mes de Nacimiento.
- Edad.
- Grado que Cursa.
- Modalidad.



- Situación Académica.
- Plan de Estudio.
- Año Lectivo.
- Fecha de Matricula.
- Asignaturas a llevar por el estudiante.

**Los datos necesarios a introducir son:**

- Código de Matricula: Es un dato obligatorio.

#### **3.1.6.1.3 Proceso**

Los datos del estudiante aparecerán al buscarlo por apellido y podrán ser actualizados. Una vez introducidos los datos se podrán almacenar en la base de datos.

#### **3.1.6.1.4 Salida.**

Se generará un reporte de matrícula del estudiante.

#### **3.1.7 Captura de los datos para el ingreso de profesores.**

##### **3.1.7.1 Especificación**

###### **3.1.7.1.1 Introducción**

Se mostrará un formulario donde se podrán introducir los datos de un profesor.

###### **3.1.7.1.2 Entradas**

**Por teclado:**

- Nombres.
- Apellidos.
- Teléfono.



- Municipio.
- Año de Nacimiento.
- Dirección de Residencia.
- Cédula.
- Observaciones.

**Por el sistema:**

- Código del Profesor.
- Sexo.
- Departamento de Nacimiento.
- Día de Nacimiento.
- Mes de Nacimiento.
- Edad.
- Status.

**Los datos necesarios a introducir son:**

- Nombres: Es un dato obligatorio.
- Apellidos: Es un dato obligatorio.
- Municipio: Es un dato obligatorio.
- Año de Nacimiento: Es un dato obligatorio.
- Dirección de Residencia: Es un dato obligatorio.
- Cédula: Es un dato obligatorio.



### **3.1.7.1.3 Proceso**

Se mostrará la pantalla de introducción de datos al usuario. El Código del Profesor y Edad aparecerán automáticamente. El Sexo, Departamento de Nacimiento, Día de Nacimiento, Mes de Nacimiento y Status podrán ser seleccionados de una lista. El resto de datos deberán ser introducidos por teclado.

Tendrá la opción de: buscar un determinado profesor introduciendo su número de cédula y modificar sus datos.

### **3.1.7.1.4 Salida.**

Se almacenará el registro del profesor en la base de datos.

## **3.1.8 Captura de los datos para el ingreso de asignatura.**

### **3.1.8.1 Especificación**

#### **3.1.8.1.1 Introducción**

Se mostrará un formulario donde se podrán introducir los datos de una asignatura.

#### **3.1.8.1.2 Entradas**

##### **Por teclado:**

- Código de Asignatura.
- Nombre de Asignatura.
- Nombre Área.

##### **Por el sistema:**

- Semestre.
- Grado.
- Código Plan de Estudio.



**Los datos necesarios a introducir son:**

- Código de Asignatura: Es un dato obligatorio.
- Nombre de Asignatura: Es un dato obligatorio.
- Nombre Área: Es un dato obligatorio.

#### **3.1.8.1.3 Proceso**

Se mostrará la pantalla de introducción de datos al usuario. El Semestre, Grado y Código Plan de Estudio podrán ser seleccionados de una lista. El resto de datos deberán ser introducidos por teclado.

Tendrá la opción de: buscar una determinada asignatura introduciendo el nombre de la asignatura y modificar sus datos.

#### **3.1.8.1.4 Salida.**

Se almacenará el registro de la asignatura en la base de datos.

### **3.1.9 Captura de los datos para el ingreso de un plan de estudio.**

#### **3.1.9.1 Especificación**

##### **3.1.9.1.1 Introducción**

Se mostrará un formulario donde se podrán introducir los datos de un plan de estudio.

##### **3.1.9.1.2 Entradas**

**Por teclado:**

- Código del Plan de Estudio.
- Nombre del Plan.
- Año.

**Por el sistema:**

- Estado.



**Los datos necesarios a introducir son:**

- Código del Plan de Estudio: Es un dato obligatorio.
- Nombre del Plan: Es un dato obligatorio.
- Año: Es un dato obligatorio.

#### **3.1.9.1.3 Proceso**

Se mostrará la pantalla de introducción de datos al usuario. El Estado podrá ser seleccionado de una lista. El resto de datos deberán ser introducidos por teclado.

Tendrá la opción de: buscar un determinado plan de estudio introduciendo el código del plan de estudio y modificar sus datos.

#### **3.1.9.1.4 Salida.**

Se almacenará el registro del plan de estudio en la base de datos.

### **3.1.10 Agregar las calificaciones de los estudiantes.**

#### **3.1.10.1 Especificación**

##### **3.1.10.1.1 Introducción**

Se mostrará un formulario donde se podrá introducir las calificaciones de un estudiante.

##### **3.1.10.1.2 Entradas**

**Por teclado:**

- IB.
- IIB.
- IIIB.
- IVB.





**Por el sistema:**

- Nivel Académico.
  
- Año.
  
- Grado.
  
- Turno.
  
- Número de Carnet.
  
- Nombres.
  
- Apellidos.
  
- Asignatura.
  
- [IS].
  
- IIS.
  
- NF.

**3.1.10.1.3 Proceso**

Se mostrará la pantalla de introducción de datos al usuario. El Nivel Académico podrá ser seleccionado de una lista, el Grado al que pertenece podrá ser seleccionado de una serie de botones de opción que oscilan de primer grado al onceavo grado, el Turno podrá ser seleccionado activando la opción correspondiente al turno al que pertenece el alumno, el Número de Carnet, Nombres, Apellidos serán seleccionados de una lista, El IP, IIP, IIIP, IVP, VP serán introducidos por teclado. Una vez introducido los datos se podrán calcular el [IS], el IIS y la NF.

**3.1.10.1.4 Salida.**

Se almacenara en la base de datos la información introducida en el formulario.



### **3.1.11 Reporte de Lista de Estudiante.**

#### **3.1.11.1 Especificación**

##### **3.1.11.1.1 Introducción**

Mostrará por pantalla el listado de estudiantes de un determinado grado, año y turno.

##### **3.1.11.1.2 Entradas**

**Por teclado:**

- Año.

**Por el sistema:**

- Grado.
- Turno.

**Los datos necesarios a introducir serán:**

- Año: es un dato obligatorio.

##### **3.1.11.1.3 Proceso**

Se mostrará por pantalla una interfaz gráfica donde el usuario del sistema podrá introducir el grado, turno y año. Se mostrará por pantalla el listado de alumnos.

##### **3.1.11.1.4 Salida.**

Se reflejará por pantalla el listado de estudiantes.

### **3.1.12 Reporte de Matrícula por Estudiante.**

#### **3.1.12.1 Especificación**

##### **3.1.12.1.1 Introducción**

Mostrará por pantalla los datos de la matrícula de un estudiante.



### **3.1.12.1.2 Entradas**

**Por pantalla:**

- Apellido.

**Los datos necesarios a introducir serán:**

- Apellido: es un dato obligatorio.

### **3.1.12.1.1 Proceso**

Se mostrará por pantalla una interfaz gráfica donde el usuario del sistema podrá Buscar al estudiante y se mostrará por pantalla su hoja de matrícula.

### **3.1.12.1.1 Salida**

Se reflejará por pantalla la hoja de matrícula del estudiante.

## **3.1.13 Reporte de Calificaciones por Estudiante**

### **3.1.13.1 Especificación**

#### **3.1.13.1.1 Introducción**

Mostrará por pantalla las calificaciones de un estudiante.

#### **3.1.13.1.2 Entradas**

**Por teclado:**

- Apellido.
- Grado.
- Año.

**Los datos necesarios a introducir serán:**

- Apellido: es un dato obligatorio.



- Grado: es un dato obligatorio.
- Año: es un dato obligatorio.

#### **3.1.13.1.3 Proceso**

Se mostrará por pantalla una interfaz gráfica donde el usuario del sistema podrá Buscar al estudiante y se mostrará por pantalla su hoja de calificaciones.

#### **3.1.13.1.4 Salida.**

Se reflejará por pantalla la hoja de calificaciones del estudiante.

### **3.1.14 Crear Copia de Seguridad**

#### **3.1.14.1 Especificación**

##### **3.1.14.1.1 Introducción**

Se podrá crear una copia de seguridad de los datos almacenados en la base de datos.

##### **3.1.14.1.2 Entradas**

**Por teclado:**

- Directorio.

**Los datos necesarios a introducir serán:**

- Directorio: es un dato obligatorio.

##### **3.1.14.1.3 Proceso**

Al dar clic en el botón Examinar se desplegará el formulario para que pueda seleccionar el directorio donde se realizará la copia de seguridad, luego presionar el botón Respaldar para poder realizar el respaldo de la base de datos.

##### **3.1.14.1.4 Salida.**

Realizar la Copia de Seguridad en el directorio seleccionado.



### **3.1.15 Restaurar copia de Seguridad.**

#### **3.1.15.1 Especificación**

##### **3.1.15.1.1 Introducción**

Se podrá restaurar una copia de seguridad de los datos almacenados en la base de datos.

##### **3.1.15.1.2 Entradas**

**Por pantalla:**

- Directorio.

**Los datos necesarios a introducir serán:**

- Directorio: es un dato obligatorio.

##### **3.1.15.1.3 Proceso**

Se presionará el botón Examinar para elegir la dirección de donde se encuentra almacenado el respaldo antes creado, luego se presionará el botón Restaurar que realizará la operación de restauración de la base de datos.

##### **3.1.15.1.4 Salida.**

Se restaurará el respaldo de la base de datos del sistema.



## X. CASOS DE USO



## X. CASOS DE USO



Figura No.3 Caso de uso Administrador

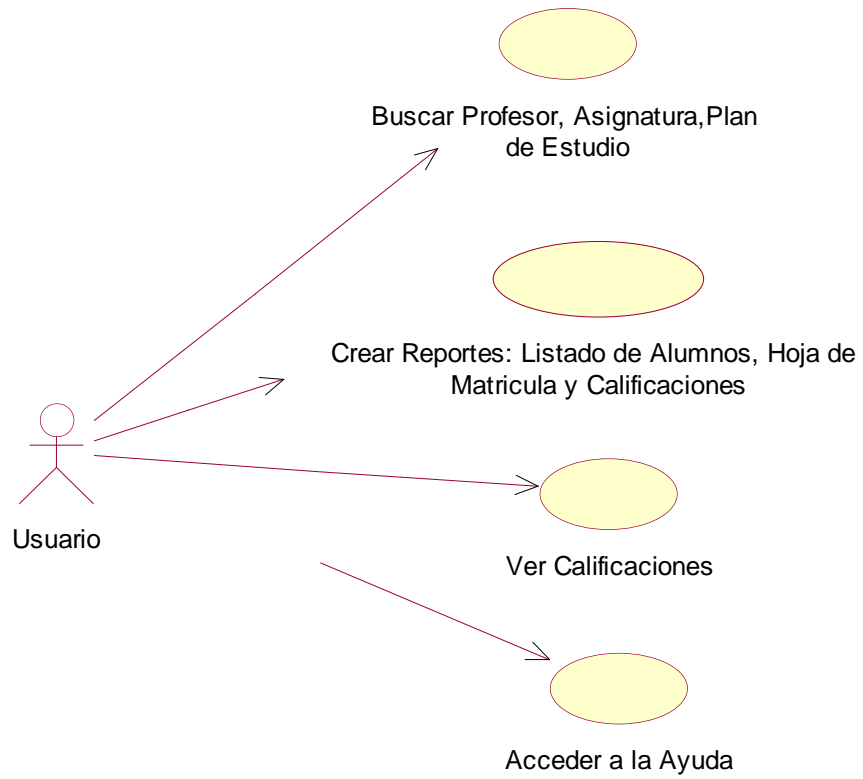


Figura No.4 Caso de Uso Usuario





*Roles de los Usuarios.*

	Matrícula	Profesor	Asignatura	Plan de Estudio	Notas	Usuario	Reporte	Copia de seguridad	Ayuda
<b>Administrador</b>	✓ Registrar.	✓ Registrar. ✓ Modificar. ✓ Visualizar. ✓ Buscar.	✓ Registrar. ✓ Modificar. ✓ Visualizar. ✓ Buscar.	✓ Registrar. ✓ Modificar. ✓ Visualizar. ✓ Buscar.	✓ Registrar. ✓ Modificar. ✓ Visualizar.	✓ Crear cuenta. ✓ Modificar Cuenta. ✓ Eliminar cuenta.	✓ Generar reporte Listado de estudiantes. ✓ Generar reporte Matrícula. ✓ Generar reporte Calificaciones.	✓ Crear Copia de Seguridad. ✓ Restaurar Copia de seguridad.	✓ Visualizar.
<b>Usuarios</b>	-	✓ Visualizar. ✓ Buscar.	✓ Visualizar. ✓ Buscar.	✓ Visualizar. ✓ Buscar.	✓ Visualizar.	-	✓ Generar reporte Listado de estudiantes. ✓ Generar reporte Matrícula. ✓ Generar reporte Calificaciones	-	✓ Visualizar.

Figura No.5 Roles de los Usuarios.



## XI. DIAGRAMA DE CLASES



## XI. DIAGRAMA DE CLASE

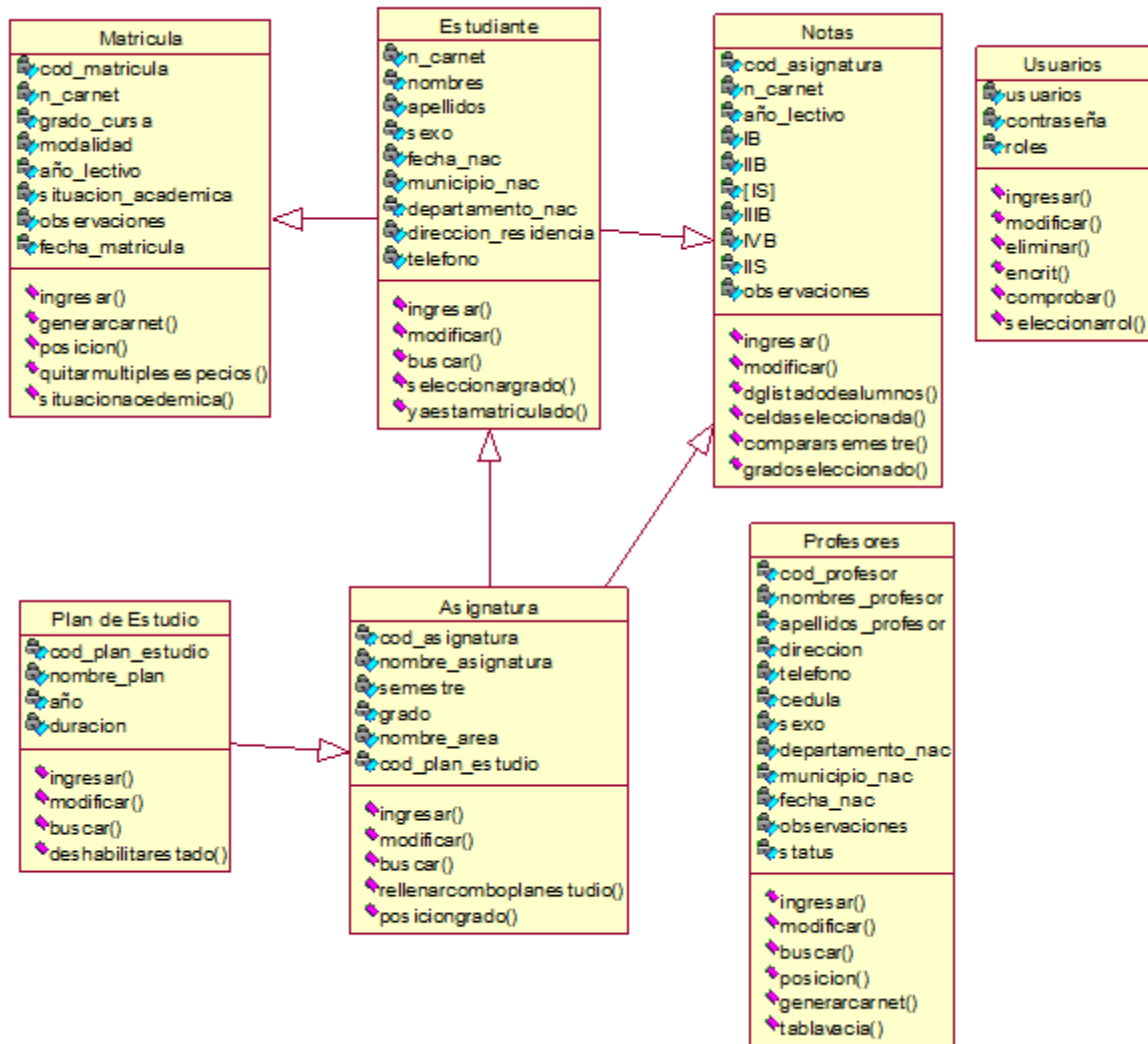


Figura No.6 Diagrama de Clase.



## XII. DIAGRAMA ENTIDAD RELACIÓN



## XII. DIAGRAMA ENTIDAD-RELACIÓN

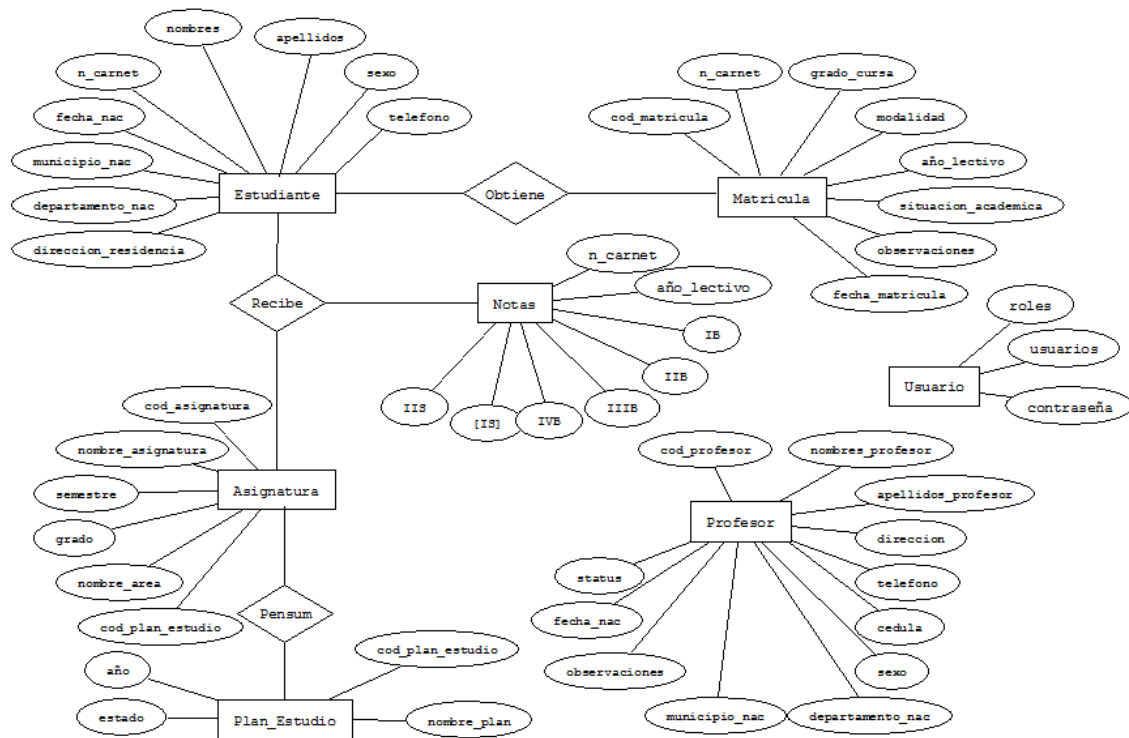


Figura No.7 Diagrama Entidad-Relación.



## XIII. FASE DE DISEÑO



### XIII. RELACIÓN DE LAS TABLAS DE LA BBDD'S

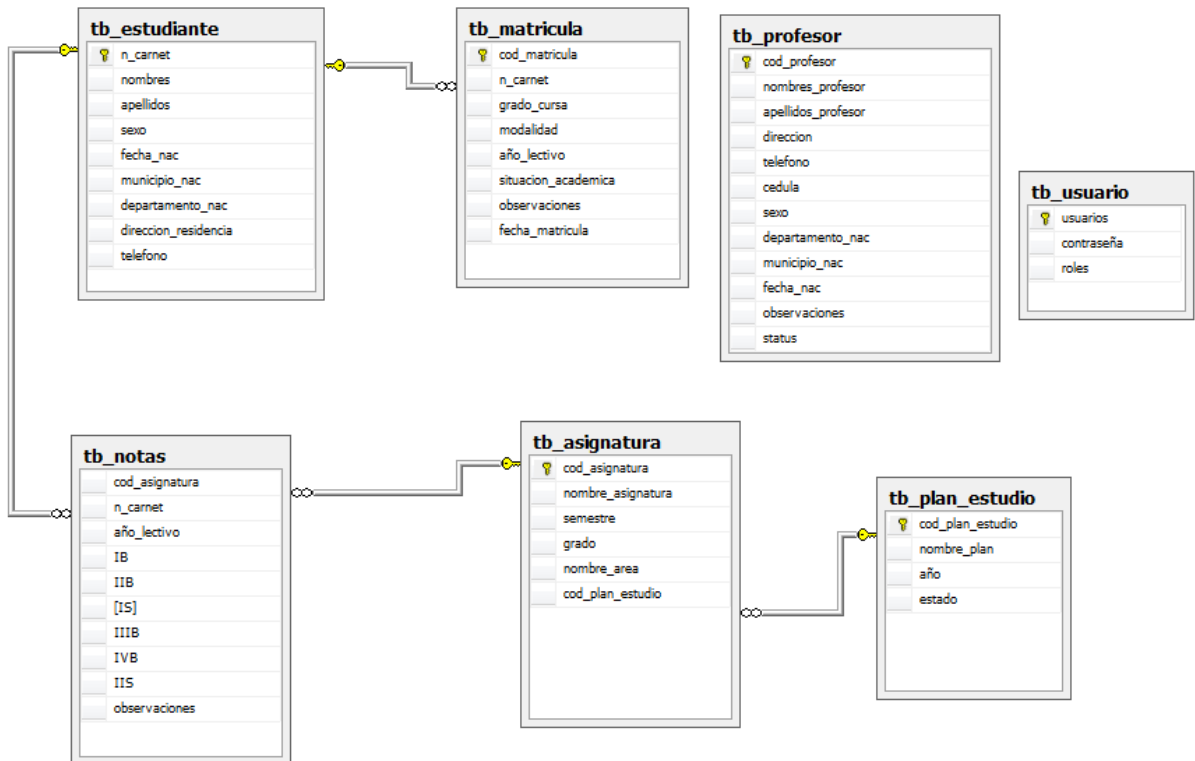


Figura No.8 Relación entre Tablas.



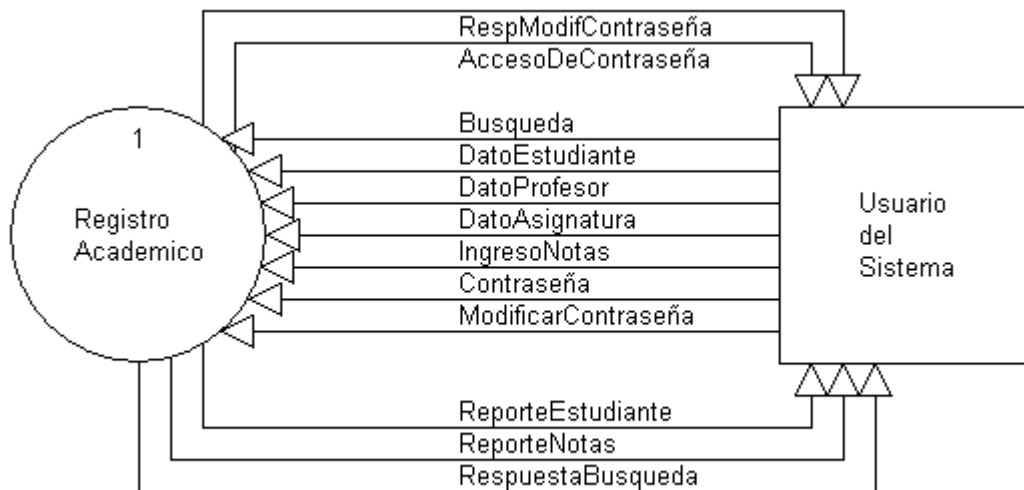
# XIV. DIAGRAMAS DE FLUJO DE DATOS



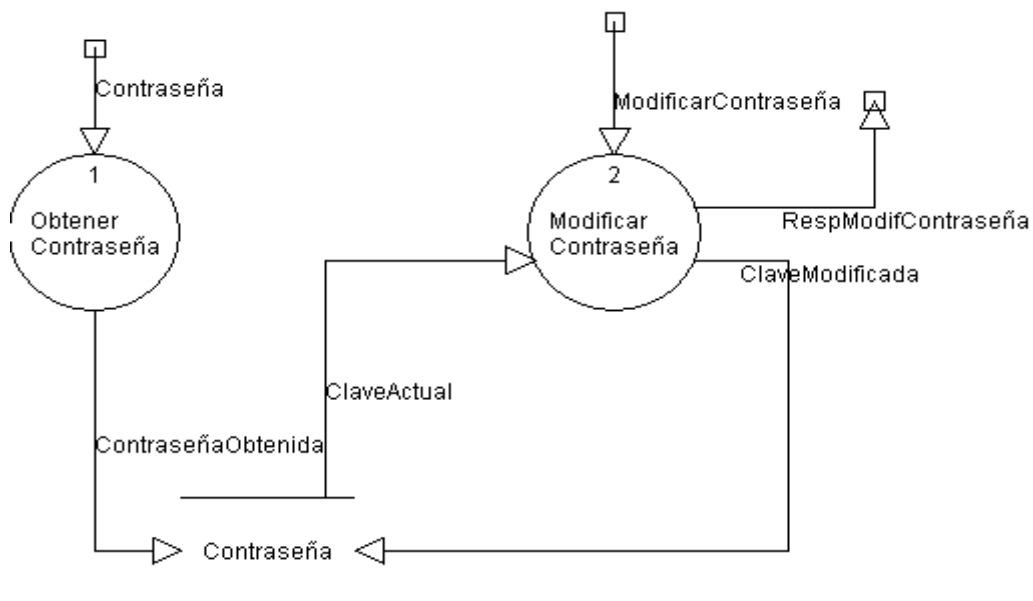


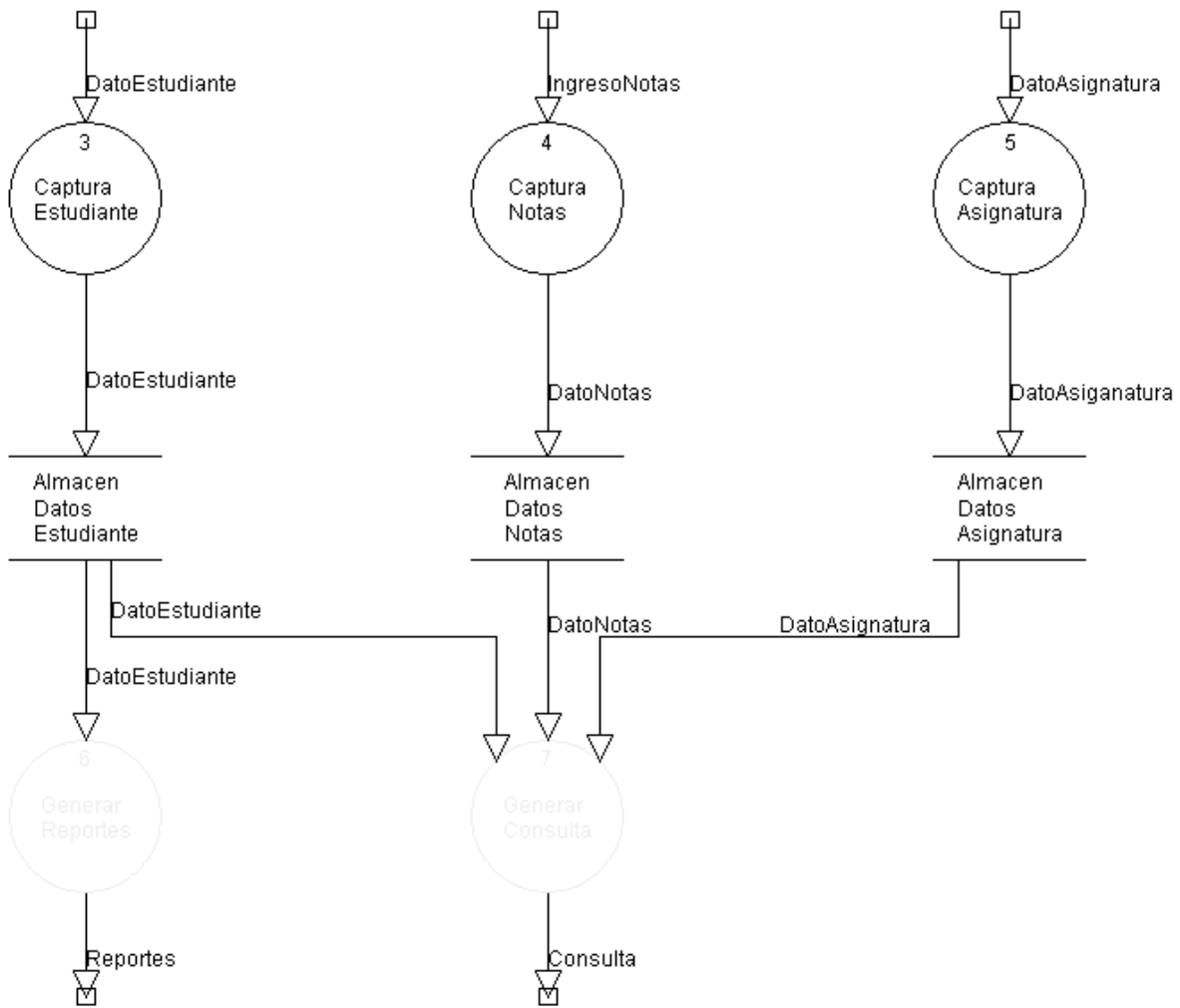
## XIV. DIAGRAMA DE FLUJOS DE DATOS

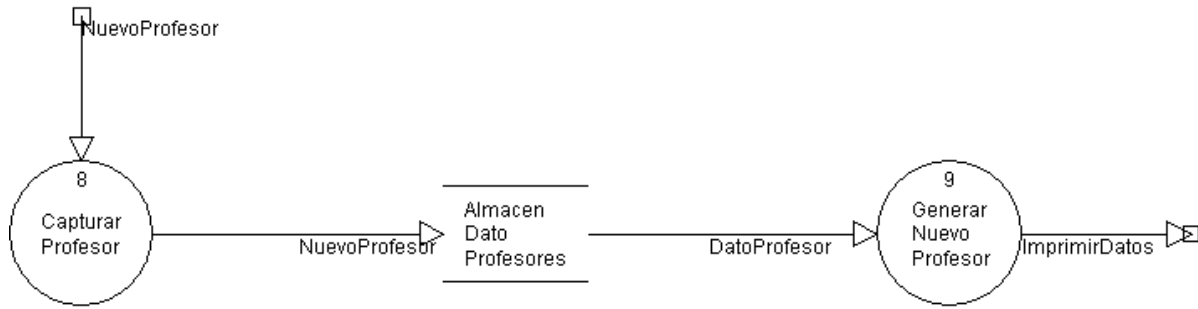
### Diagrama de Contexto

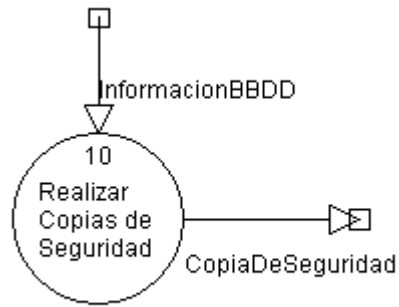


### Nivel 1

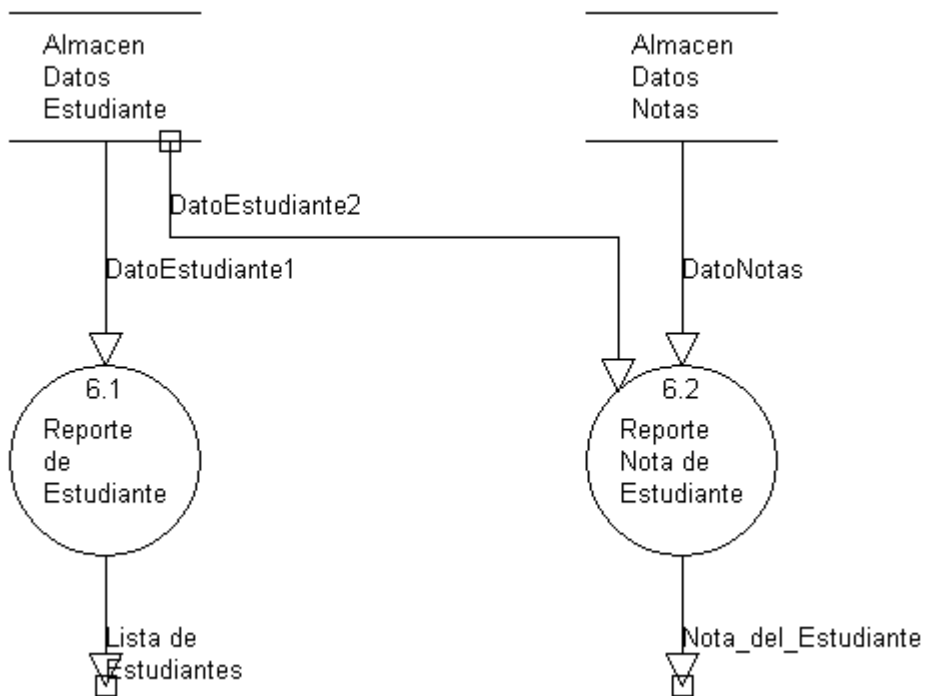


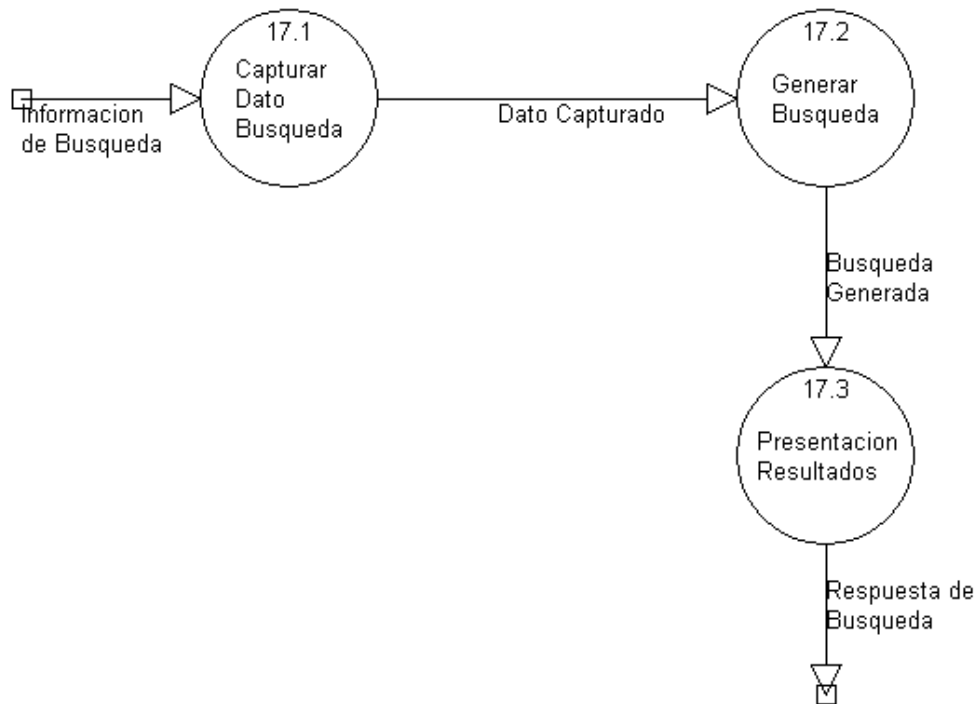
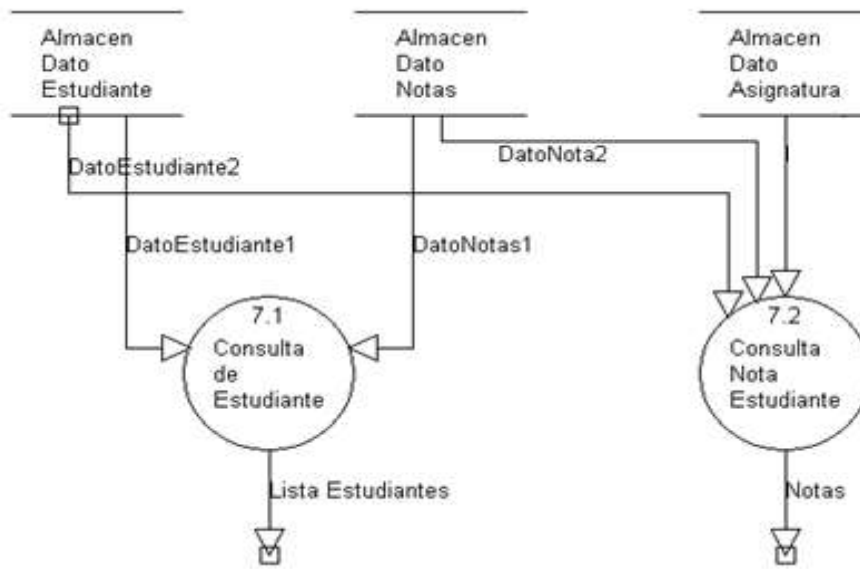






*Nivel 2*







## XV. DICCIONARIO DE DATOS



## XV. DICCIONARIO DE DATOS

**Usuario del Sistema:** Entidad externa.

**Obtener Contraseña:** Este proceso permite ingresar una contraseña.

**Contraseña:** usuario + contraseña + confirmar\_contraseña.

**Modificar Contraseña:** Este proceso permite modificar las contraseñas.

**Modificar Contraseña:** usuario + contraseña + confirmar\_contraseña.

**Captura de Estudiante:** Este proceso se permite ingresar los datos necesarios para agregar un nuevo estudiante.

**Dato Estudiante:** n\_carnet + nombres + apellidos + sexo + fecha\_nac + municipio\_nac + departamento\_nac + dirección\_residencia + teléfono.

Se utiliza para la entrada en el proceso de captura de estudiante.

**Captura Notas:** Este proceso permite registrar las notas del estudiante.

**Ingreso Notas:** n\_carnet + año\_lectivo + grado+ modalidad+IB + IIB + IIIB + IVB.

**Captura Asignatura:** Este proceso permite registrar una asignatura.

**Datos Asignatura:** cod\_asignatura + nombre\_asignatura + semestre + grado + nombre\_area + cod\_plan\_estudio.

**Generar Reportes:** Este proceso permite generar reporte de estudiantes.

**Reportes:** Se utiliza como salida en el proceso de generar reportes.

**Generar Consulta:** Este proceso permite generar consultas.

**Consultas:** Se utiliza como salida del proceso de generar consulta.

**Captura Profesor:** Este proceso permite registrar un nuevo profesor.



**Nuevo Profesor:** cod\_profesor + nombre\_profesor + apellidos\_profesor + dirección + teléfono + cedula + observaciones + status.

**Generar Nuevo Profesor:** Este proceso ingresa un nuevo profesor.

**Realizar Copias de Seguridad:** Este proceso permite realizar copias de seguridad.

**Información:** Ingresa todos los datos necesarios para realizar una copia de seguridad en la base de datos del sistema.

**CopiaDeSeguridad:** Se utiliza como salida del proceso de Realizar Copias de Seguridad.

**Reporte de Estudiante:** Este proceso permite crear un reporte de un estudiante.

**Lista de Estudiantes:** Es la salida del proceso de crear un reporte del estudiante.

**Reporte de Nota de Estudiante:** Este proceso permite crear un reporte de la nota del estudiante.

**Nota\_del\_Estudiante:** Es la salida del proceso de crear un reporte de estudiante por notas generando un boletín.

**Consulta de Estudiante:** Este proceso permite consultar datos del estudiante.

**Lista Estudiante:** Es la salida del proceso de consultar un estudiante crea un reporte con la lista de estudiantes por grado, modalidad y año lectivo.

**Consulta Nota de Estudiante:** Este proceso permite consultar la nota del estudiante.

**Notas:** Es la salida del proceso de Consulta Nota Estudiante genera un reporte con la nota del estudiante.

**Información de Búsqueda:** Se usa como entrada del proceso de Capturar Dato Búsqueda proporciona la información de la búsqueda.

**Capturar Dato Búsqueda:** Proceso que permite capturar la información de la búsqueda.

**Dato Capturado:** apellidos.

**Generar Búsqueda:** Proceso que genera la búsqueda de un estudiante por apellidos.





**Búsqueda Generada:** Salida del proceso de Generar Búsqueda y entrada del proceso de Presentación de Resultados.

**Presentación Resultados:** Proceso que permite mostrar el resultado de una búsqueda iniciada de un estudiante de reingreso que se mostrará en el formulario de matrícula.

**Respuesta de Búsqueda:** Salida del proceso de presentación Resultados que mostrará el resultado en el formulario de la matrícula del estudiante de reingreso.



## XVI. DISEÑO DE DATOS



## XVI. DISEÑO DE DATOS

*Tabla 1 Matricula*

Nombre del Campo	Tipo de Dato	Longitud	Descripción
cod_matricula	Char	10	Código asignado a cada matricula
n_carnet	char	10	Número de carnet asignado a cada estudiante
grado_cursa	Varchar	4	Gado que cursa un estudiante
modalidad	Varchar	50	Modalidad que cursa un estudiante
año_lectivo	int		Año actual que cursa un estudiante
situación_academica	Varchar	50	Estado de un estudiante con el grado que cursa
Observaciones	Varchar	MAX	Observaciones que un estudiante puede tener
fecha_matricula	Varchar	11	Fecha en la que un estudiante se matricula

*Tabla 2 Tabla Estudiante.*

Nombre del Campo	Tipo de Dato	Longitud	Descripción
n_carnet	Char	10	Número de carnet de estudiante
nombres	Varchar	50	Nombre de cada estudiante
apellidos	Nvarchar	50	Apellido de cada estudiante
sexo	Char	1	Sexo de cada estudiante
fecha_nac	Datetime		Fecha de nacimiento de los estudiantes
municipio_nac	Char	15	Municipio de nacimiento del estudiante.
departamento_nac	Char	15	Departamento de nacimiento del estudiante
direccion_residencia	Varchar	MAX	Dirección de la residencia del estudiante
telefono	Char	9	Teléfono del estudiante



*Tabla 3 Tabla Profesor*

Nombre del Campo	Tipo de Dato	Longitud	Descripción
cod_profesor	Varchar	9	Código de un profesor
nombres_profesor	Varchar	100	Nombre del profesor
apellidos_profesor	Varchar	100	Apellido del profesor
direccion	Varchar	MAX	Dirección del profesor
cedula	Varchar	16	Cedula de identidad de cada profesor
telefono	Varchar	9	Teléfono del profesor
sexo	Char	1	Indica el sexo del profesor
departamento_nac	Varchar	50	Departamento de nacimiento del profesor
municipio_nac	Varchar	50	Municipio de nacimiento del profesor
fecha_nac	Datetime		Fecha de nacimiento del profesor
observaciones	Varchar	MAX	Campo para realizar alguna observación
status	Bit		Estado actual del profesor

*Tabla 4 Tabla Asignatura*

Nombre del Campo	Tipo de Dato	Longitud	Descripción
cod_asignatura	Varchar	10	Código de asignatura
nombre_asignatura	Varchar	50	Nombre de asignatura
semestre	Varchar	4	Semestre en que se impartirá la asignatura
grado	Varchar	4	Grado que se imparte la asignatura
nombre_area	Varchar	50	Nombre del área a la que pertenece la asignatura
cod_plan_estudio	Varchar	10	Código del plan de estudio al que pertenece la asignatura

*Tabla 5 Tabla Plan de Estudio*

Nombre del Campo	Tipo de Dato	Longitud	Descripción
cod_plan_estudio	Varchar	10	Código del plan de estudio al que pertenece la asignatura
nombre_plan	Varchar	MAX	Nombre del plan de estudio
año	Varchar	4	Año de plan de estudio
estado	bit		Estado del plan de estudio

*Tabla 6 Tabla Notas*

Nombre del Campo	Tipo de Dato	Longitud	Descripción
cod_asignatura	Varchar	10	Código de asignatura
n_carnet	Char	10	Carnet del estudiante
año_lectivo	Numeric	(18,0)	Año actual
IB	Int		I Bimestre
IIB	Int		II Bimestre
[IS]	Int		I Semestre
IIIB	Int		III Bimestre
IVB	Int		IV Bimestre
IIS	Int		II Semestre
observaciones	Varchar	MAX	Campo para realizar una observación

*Tabla 7 Tabla para Contraseña de Usuarios*

Nombre del Campo	Tipo de Dato	Longitud	Descripción
usuarios	Varchar	50	Usuario del sistema
contraseña	Varchar	50	Contraseña del usuario
roles	Varchar	20	Identifica el Rol de Usuario.

## XVII. DISEÑO PROCEDIMENTAL





## XVII. DISEÑO PROCEDIMENTAL

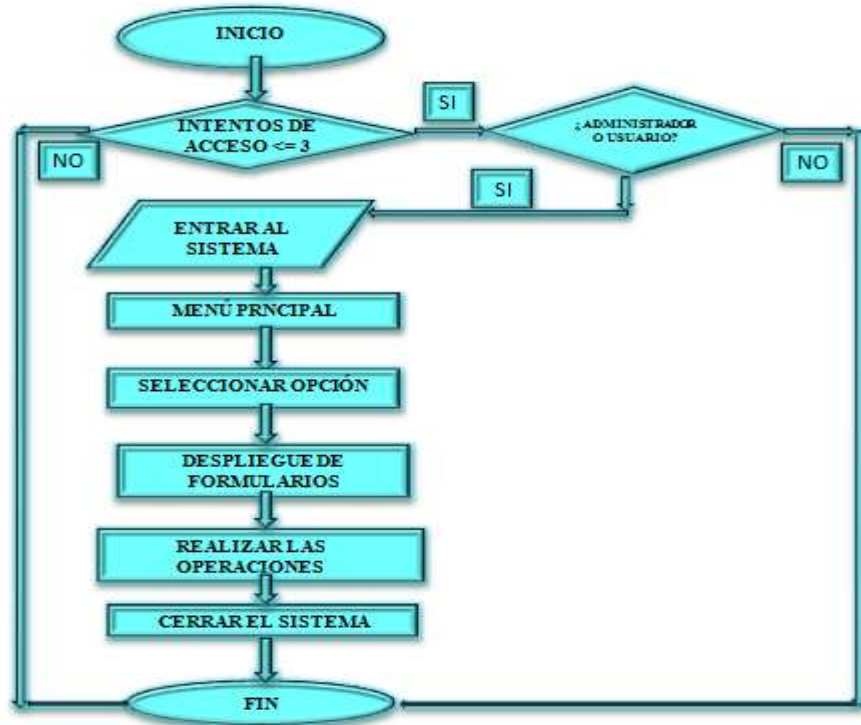


Figura No.9 Diseño Procedimental.



# XVIII. DISEÑO ARQUITECTÓNICO



## XVIII. DISEÑO ARQUITECTÓNICO

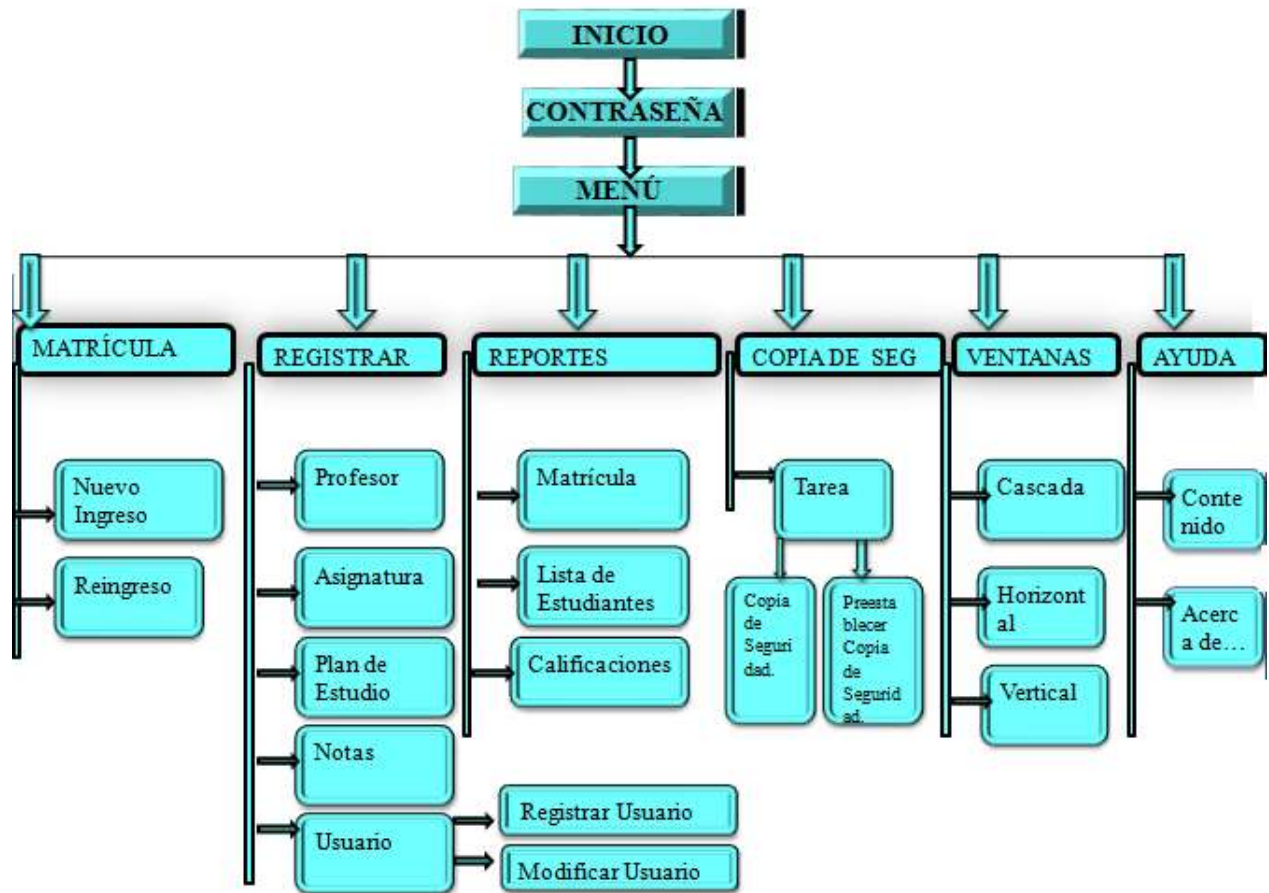


Figura No.10 Diseño Arquitectónico.



# XIX. DESARROLLO DE FORMULARIOS



## XIX. DISEÑO DE INTERFACES

El sistema constara de las siguientes interfaces (formularios):

*Ventana de Presentación.*

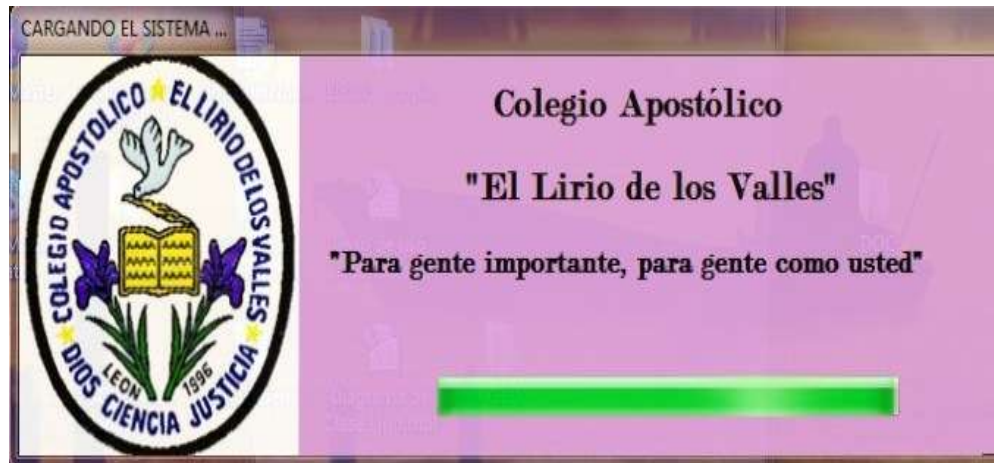


Figura No.11 Presentación.

*Ventana de inicio de Sesión*



Figura No.12 Inicio de Sesión.



### Ventana Principal



Figura No.13 Menú Principal.

### Registro de Estudiantes.

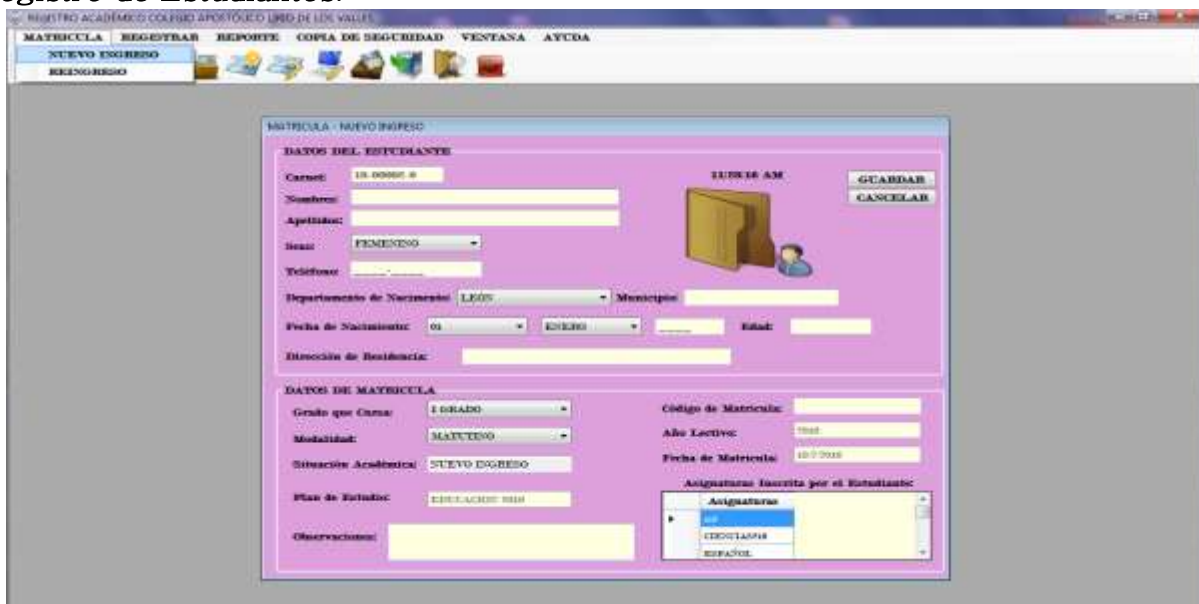


Figura No.14 Registro de un Estudiante de Nuevo Ingreso.



*Búsqueda de Estudiante para un Reingreso.*

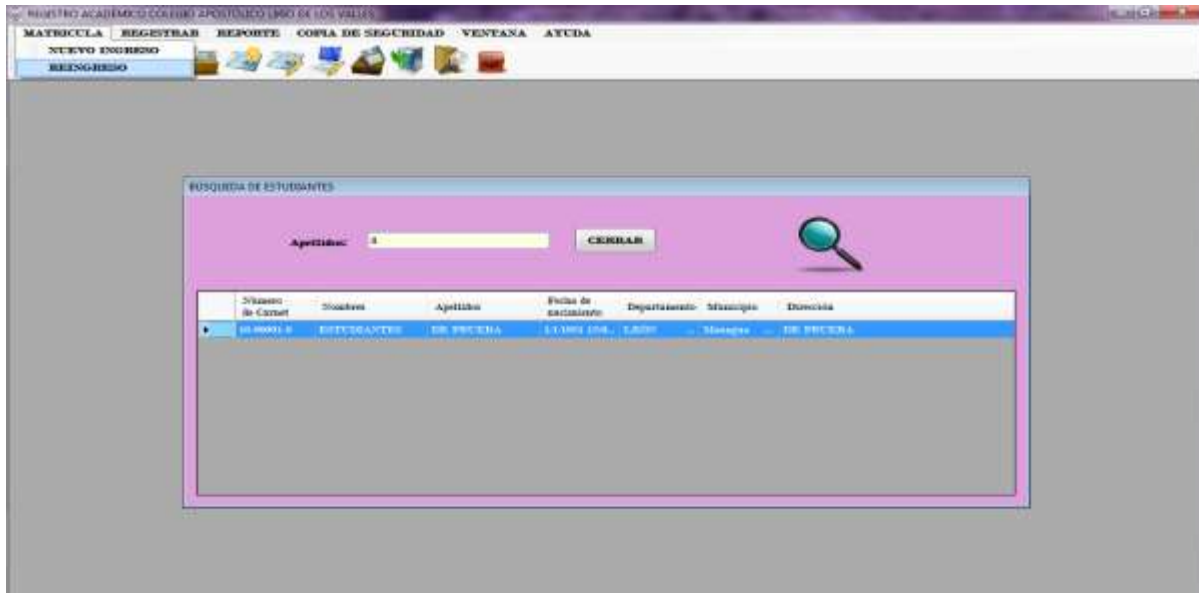


Figura No.15 Estudiante de Reingreso.

*Registrar Profesor.*

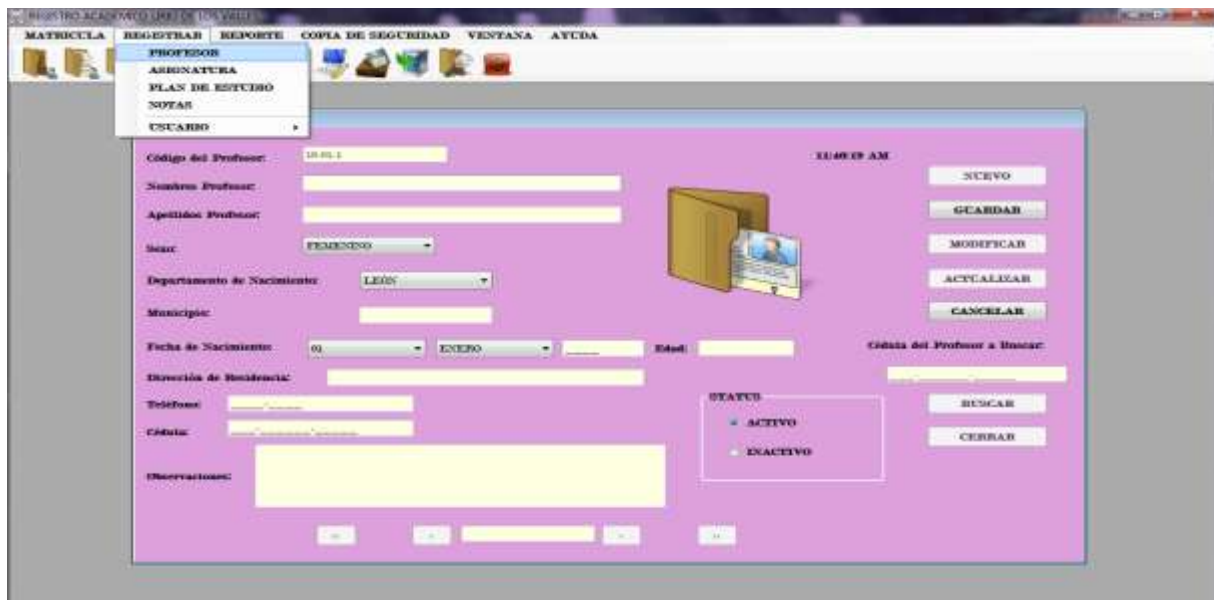


Figura No.16 Registro de Profesor.



*Ingresar Asignatura*

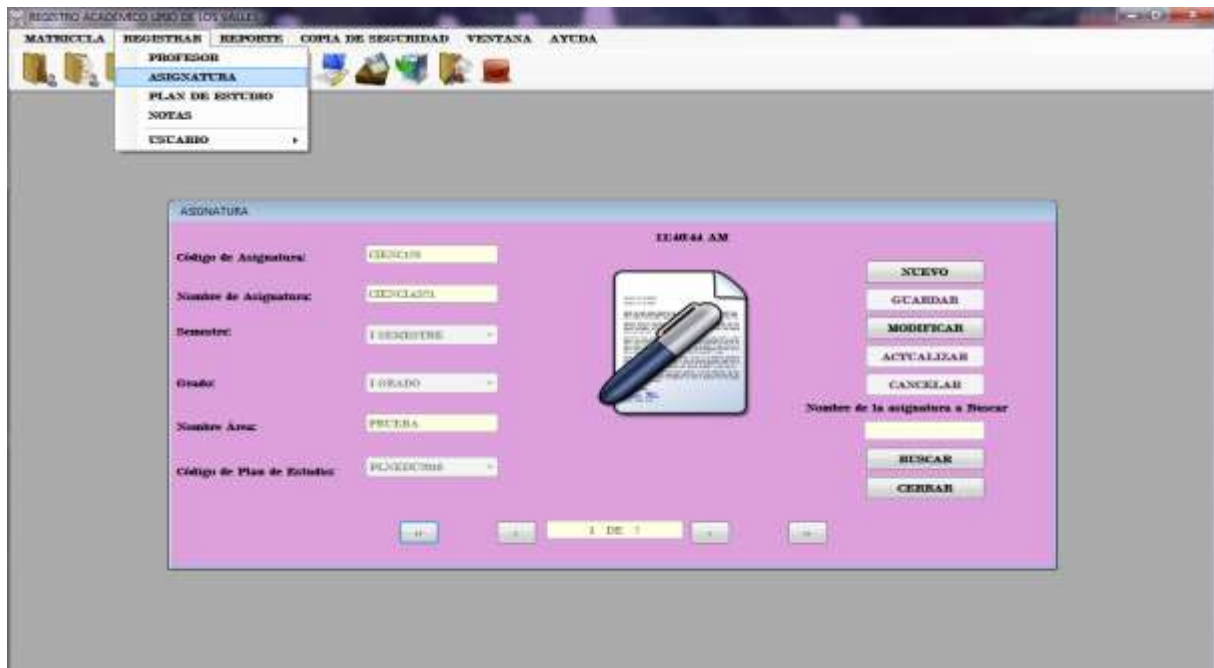


Figura No.17 Registro de Asignatura.

*Ingreso de Plan de Estudio.*

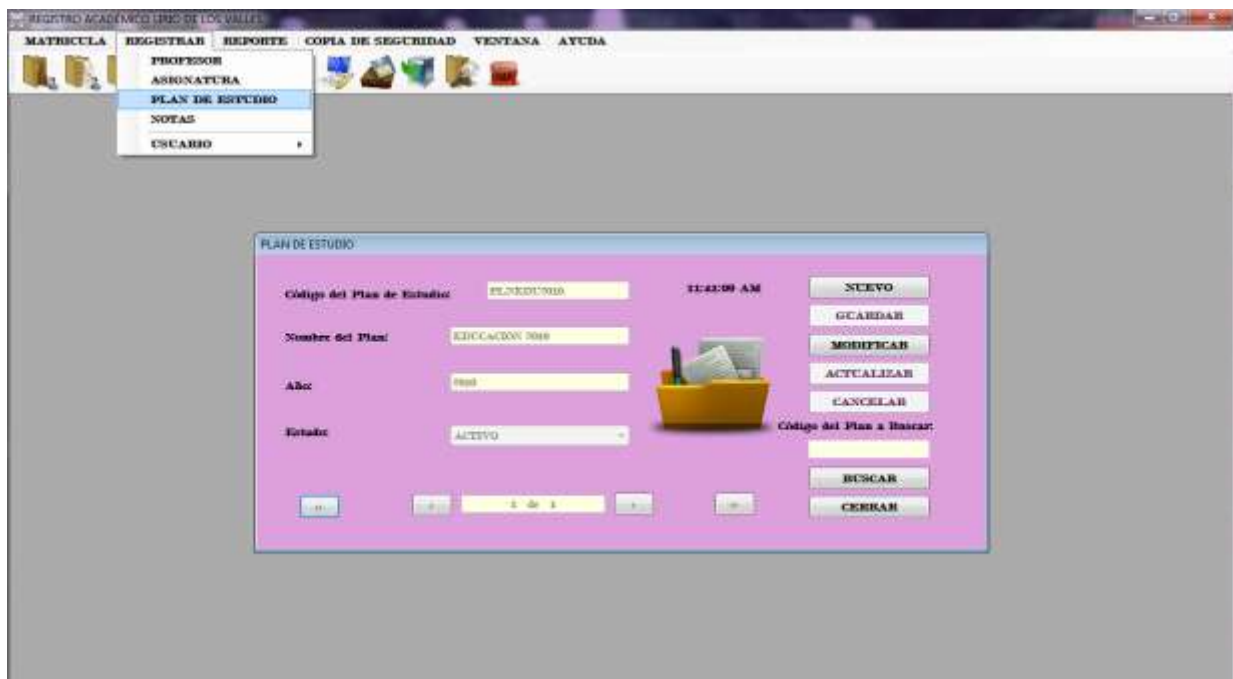


Figura No.18 Registro Plan de Estudio.





### Registrar Usuario



Figura No.19 Registro de Usuario.

### Realizar Copia de Seguridad.

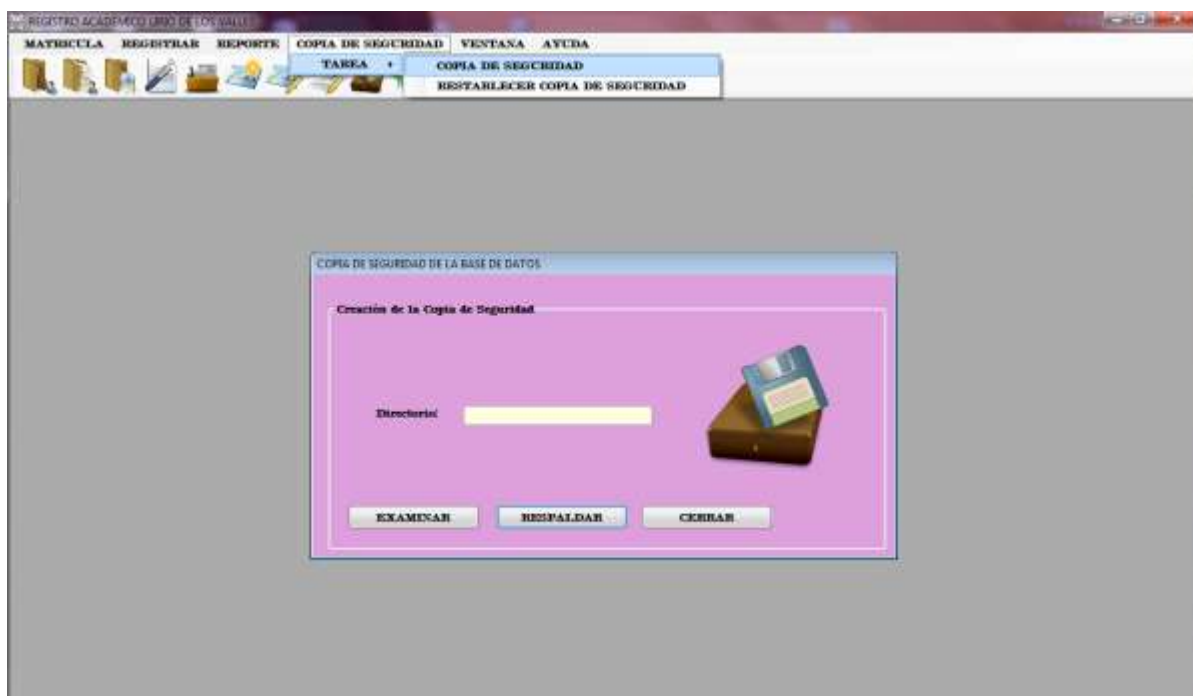


Figura No.20 Realizar Copia de Seguridad.



*Restaurar Copia de Seguridad.*

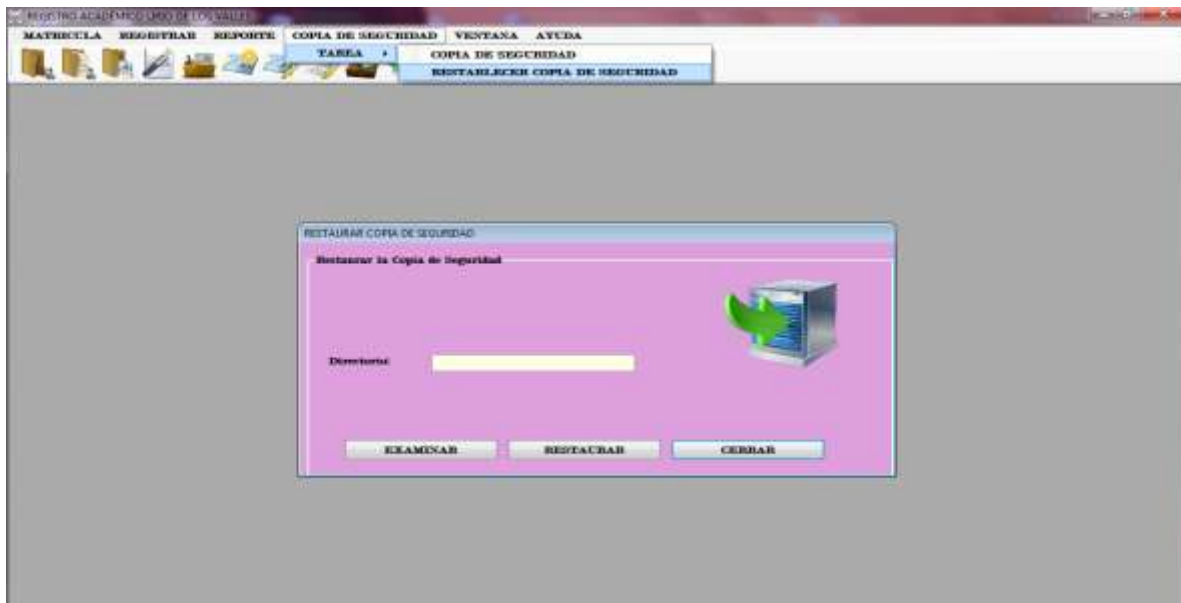


Figura No.21 Restaurar Copia de Seguridad.

*Formulario Acerca de...*

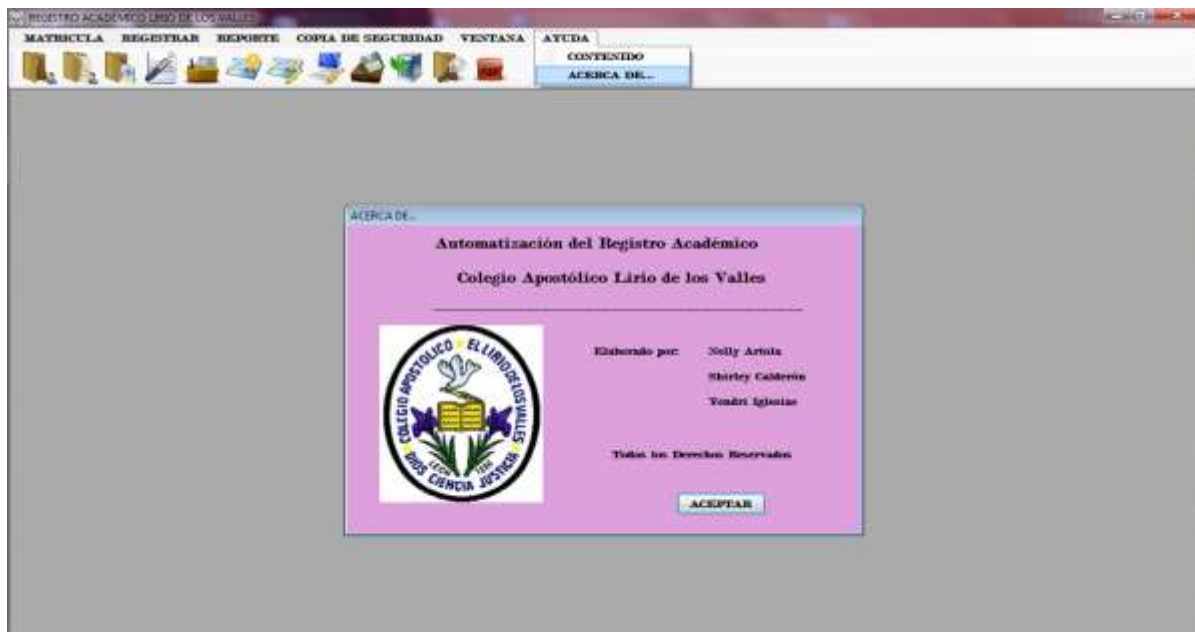


Figura No.22 Acerca de.



*Formulario de Notas.*

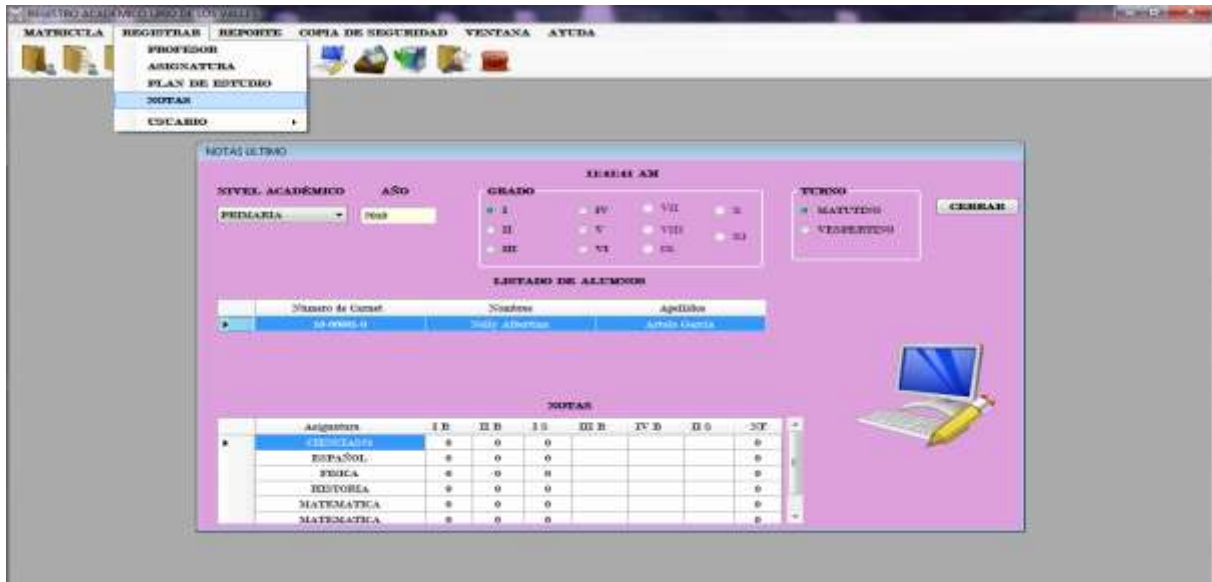


Figura No.23 Registro de Notas.

*Formulario para creación de Boletines.*

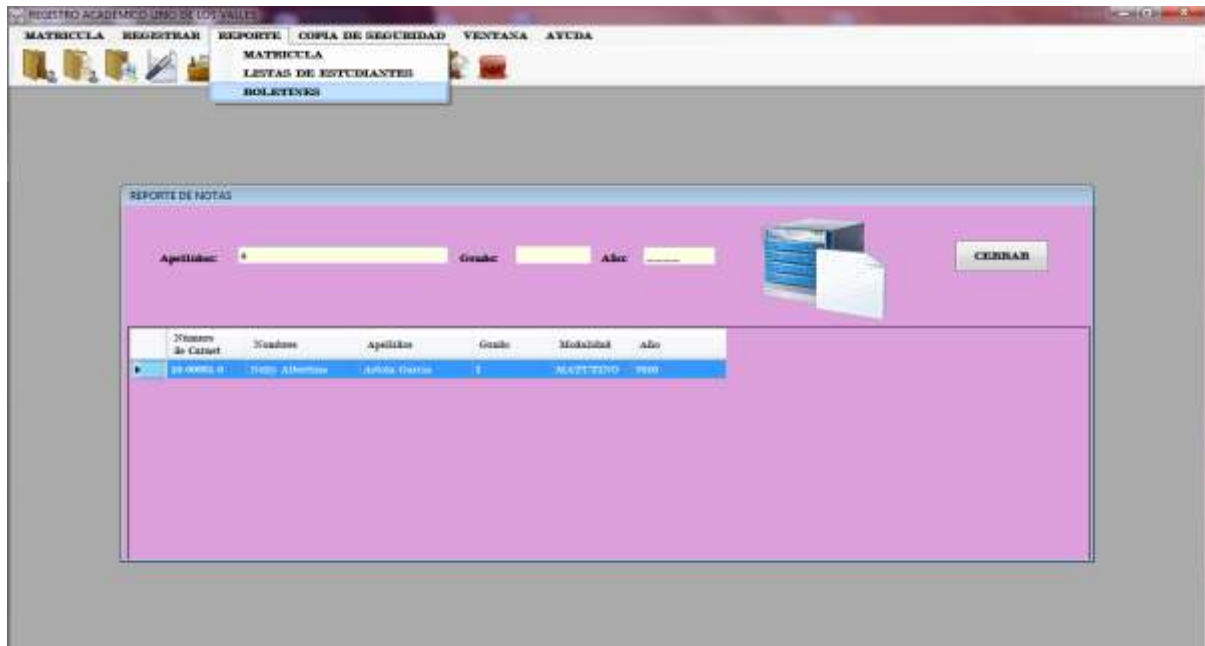


Figura No.24 Introducción de Datos para generar Certificados de Calificaciones.



*Reporte de Notas.*

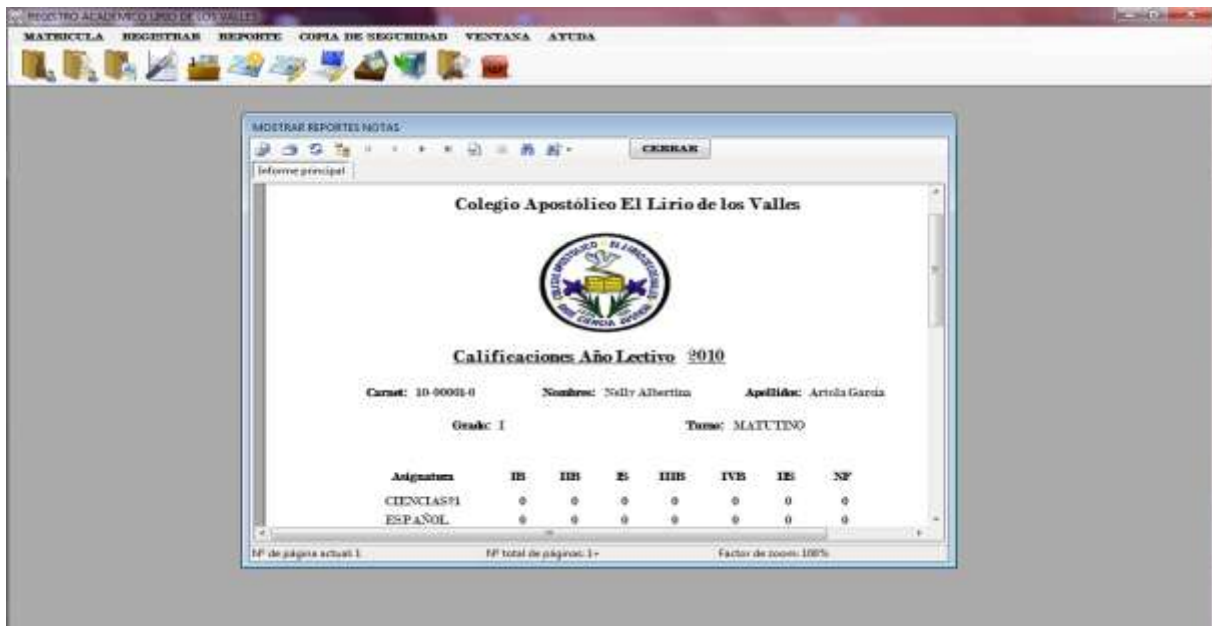


Figura No.25 Reporte de Calificaciones.

*Formulario para creación de Listado de Estudiantes.*

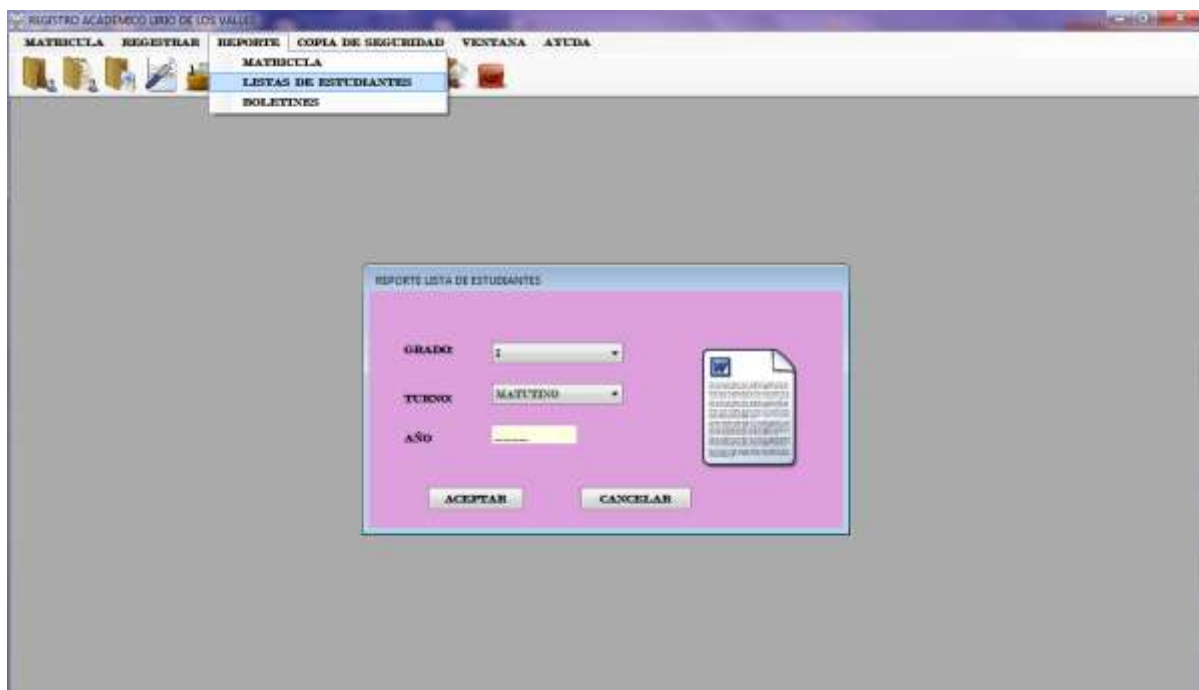
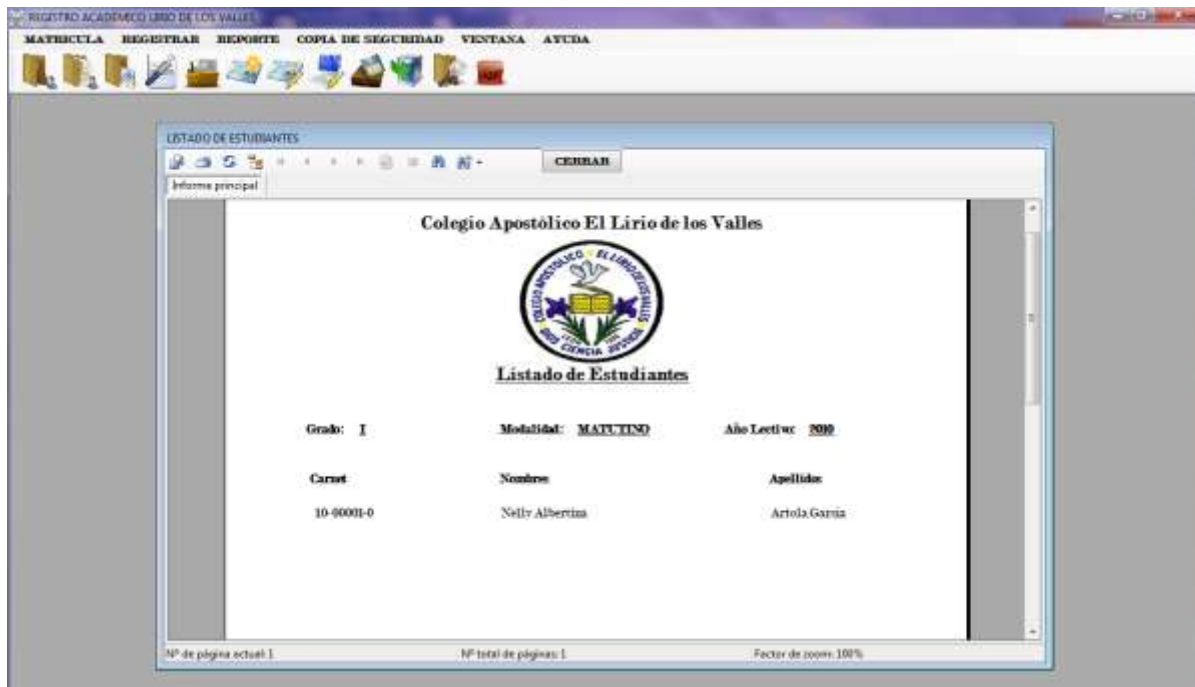


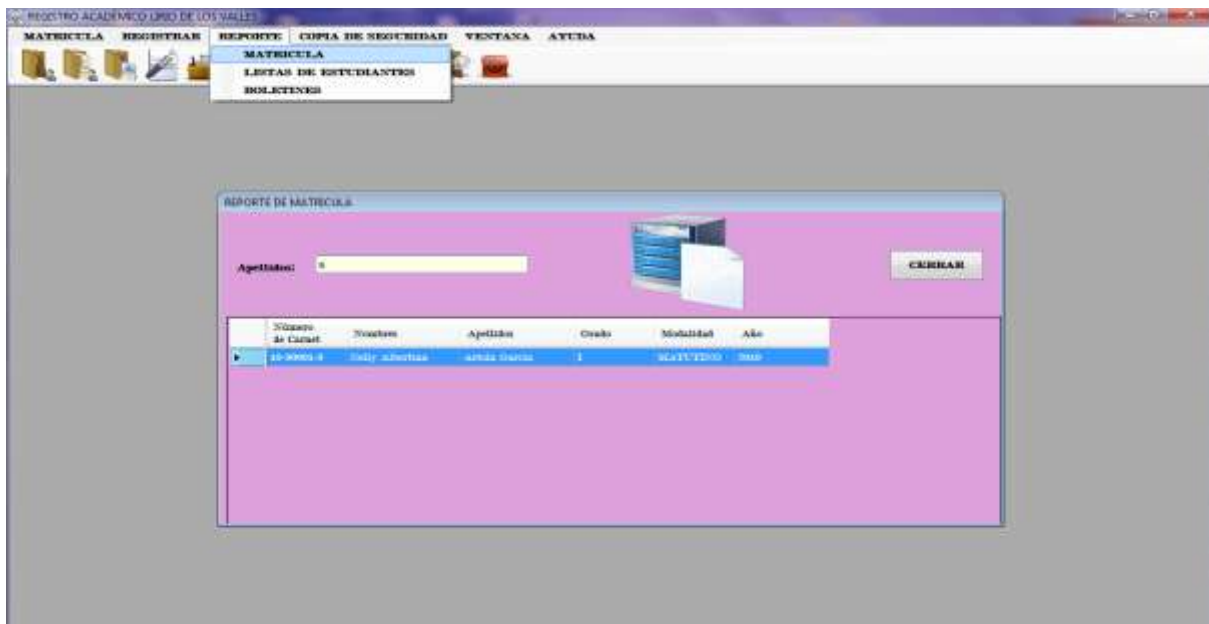
Figura No.26 Introducción de Datos para obtener Listado de Estudiantes.



*Reporte de Listado de Estudiante.*



*Formulario para creación de una hoja de Matricula.*





*Reporte de Matrícula.*

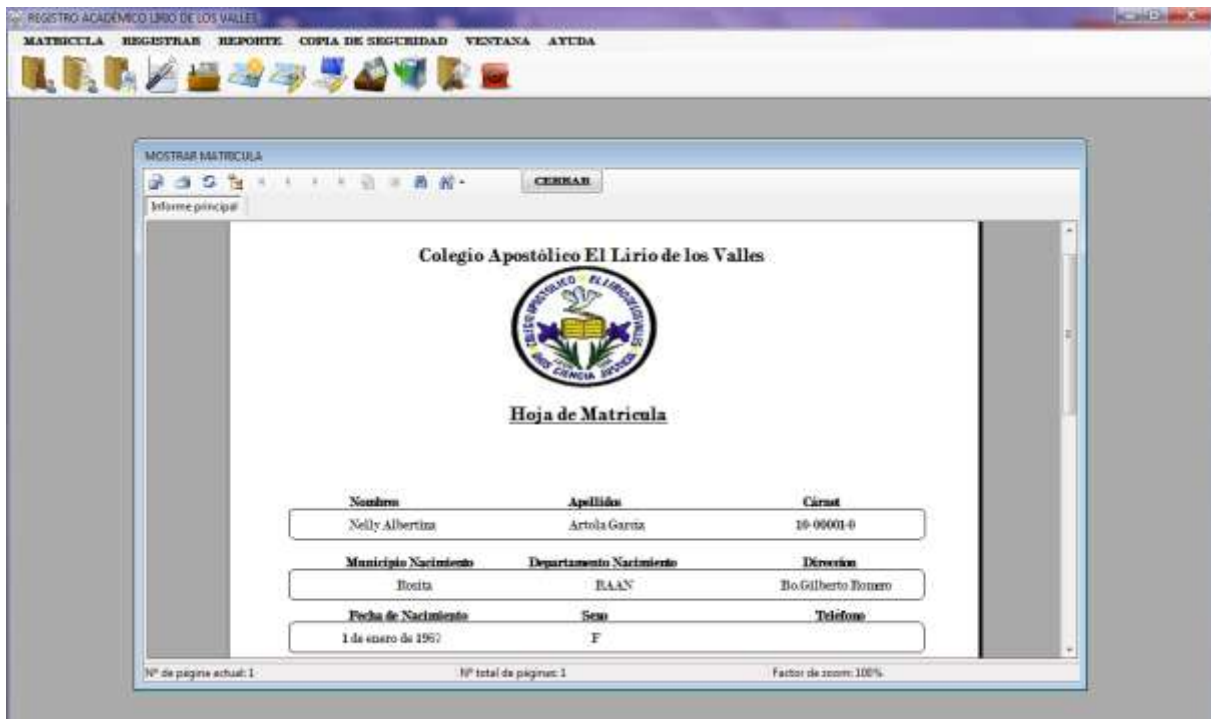


Figura No.29 Reporte de matrícula.



## XX. CONCLUSIÓN



## XX. CONCLUSIÓN

Al finalizar nuestro trabajo monográfico, hemos podido lograr los objetivos que nos habíamos planteados inicialmente, realizando de esta manera un software que cumple los requerimientos del Colegio Apostólico Lirio de los Valles y que satisface las necesidades planteadas al iniciar la creación de nuestro sistema de registro académico.

Con la elaboración de este sistema se ayudará al colegio Apostólico Lirio de los Valles a realizar un manejo rápido y mayor control de la información de estudiantes y docentes que en este centro se encuentran actualmente, generando reportes y listados en el momento que los administradores del sistema así lo deseen.

Al realizar nuestro software utilizamos Visual Studio.NET 2005 y llegamos a la conclusión que es una plataforma de mucho potencial y que incluye herramientas que permiten al programador crear aplicaciones de muy buena calidad trabajando en un entorno fácil, confiable y seguro para la elaboración de aplicaciones y en sí de nuestro sistema.





## XXI. RECOMENDACIONES



## XXI. RECOMENDACIONES

Para utilizar el sistema es necesario que los usuarios tengan conocimientos previos para poder realizar la manipulación de forma segura de la información.

Se podrá utilizar el sistema solo con la autorización respectiva del director del Colegio Apostólico Lirio de los Valles que será quien realizará las contraseñas para los usuarios del sistema.

El ordenador en que se va a instalar el sistema debe de cumplir con los requerimientos antes establecidos, para lograr el buen funcionamiento del sistema.

En el ordenador que se instalará el software el sistema debe tener actualizada la fecha y el formato de fecha deberá ser dd/MM/aaaa.

No borrar de forma directa la información almacenada en la tabla usuarios, porque si no perdemos al administrador por defecto de toda la aplicación.

Como trabajo futuro recomendamos que el acceso a la base de datos de la aplicación se realice a través de un *Login* y una *Password*. Para ello se debe hacer uso del tipo de autenticación: “*SQL Server Authentication*”. Esto añadirá un mayor nivel de seguridad sobre la base de datos de la aplicación.



## XXII. BIBLIOGRAFÍA.



## XXII. BIBLIOGRAFÍA

### *Libros consultados:*

- Ceballos Francisco Javier “Enciclopedia de Microsoft Visual C#”.
- Blanco, Luis Miguel “Programación en Visual Basic.Net”, Grupo EIDOS Consultaría y Documentación Informática, S.L., 2002
- Jeff Ferguson, Brian Patterson, Jason Beres, Pierre Boutquin y Meeta Gupta “La Biblia de C#” Editorial ANAYA.
- Faci, Santiago “Visual Basic.Net: Acceso a Datos”.
- Luis Miguel Blanco “Crystal Reports para Visual Studio.Net(Visual Basic.Net)”

### *Trabajos monográficos consultados:*

- Rebeca Nohemí Ruiz Ortiz, Ana Fabiola Rodríguez Pinell. “Sistema de Automatización de Registro Académico”. Julio 2008.
- Melvin Eduardo Burgalin Castro, Deybis Benossky Flores López, Jorge Luis Rodríguez. “Gestión de Automatización del Sistema de Inventario de la Farmacia Eben-Ezer de la Mina el Limón.” (GASFAR) León, Septiembre 2008.

### *Referencias de Internet:*

- [http://es.wikipedia.org/wiki/Base de Datos.](http://es.wikipedia.org/wiki/Base_de_Datos)
- <http://msdn.microsoft.com/es-es/vstudio/default.aspx>
- [http://es.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](http://es.wikipedia.org/wiki/Microsoft_Visual_Studio)
- <http://www.vtc.com/products/Microsoft-Visual-Studio-.NET-Espanol-tutorials.htm>
- [http://TutorialCrystalReports/Colabora\\_NET/InformesCrystal Reports.htm](http://TutorialCrystalReports/Colabora_NET/InformesCrystal Reports.htm)



## XXIII. ANEXOS.




## XXIII. ANEXOS

### *Creación de la Base de Datos en Microsoft SQL Server Management Express.*

Para crear una base de datos en Microsoft SQL Server Management Express debes de haber instalado el programa Microsoft SQL Server 2005.

Para la elaboración de unas bases de datos en Microsoft SQL Server Management Express siga los siguientes pasos:

1. Dirigirse a INICIO  de la pantalla principal de Windows y seleccione todos los programas, luego seleccione Microsoft SQL Server 2005 luego SQL Server Management Studio Express como se muestra en el grafico.

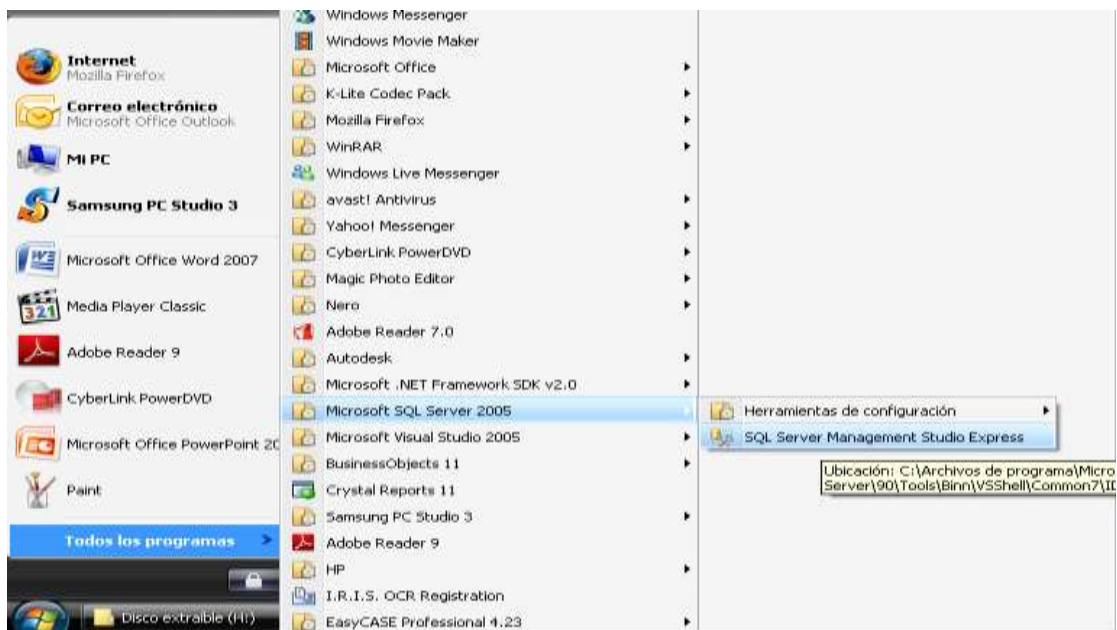


Figura No.30 Microsoft SQL Server 2005.



2. Escriba el nombre del servidor y presione el botón Conectar. 

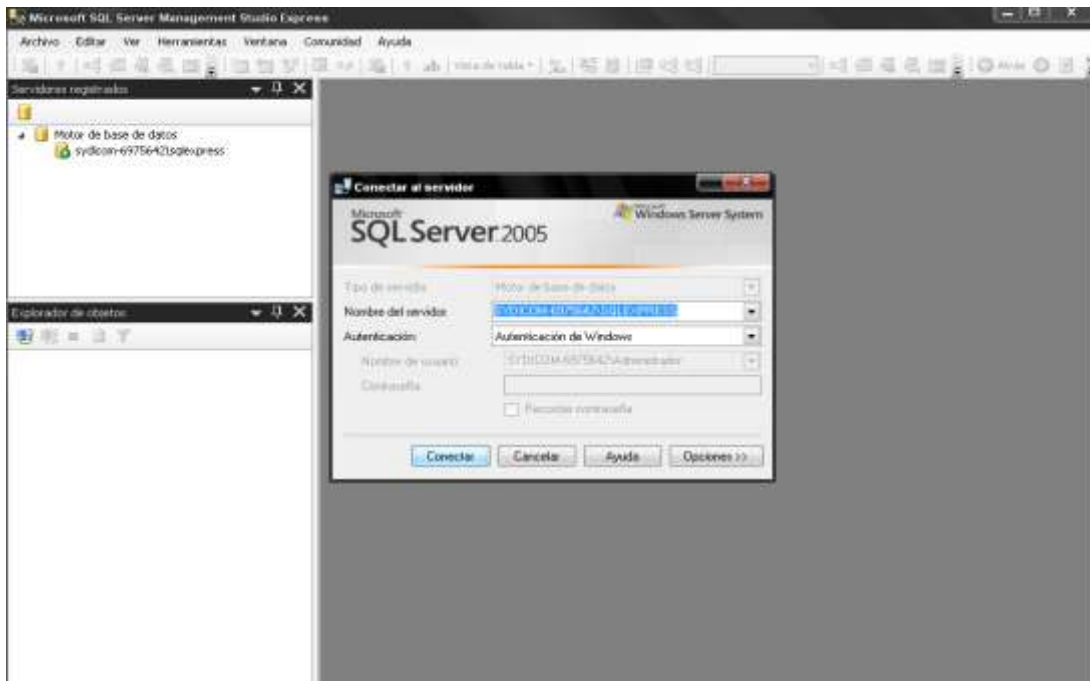


Figura No.31 Conectarse a Microsoft SQL Server 2005.

3. Al estar conectado como se muestra en la figura se podrá acceder a la creación de la base de datos.

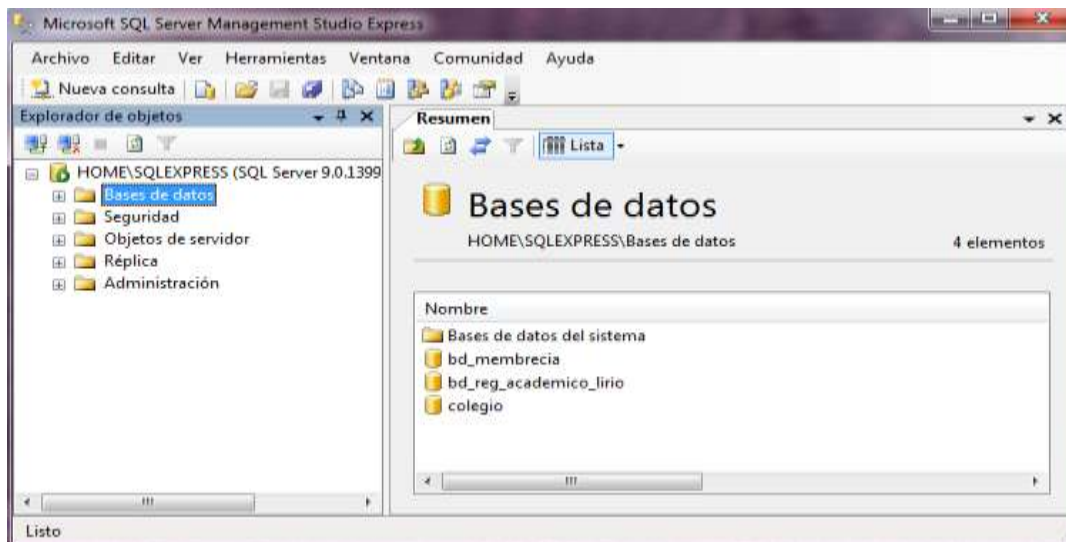


Figura No.32 Entorno de Microsoft SQL Server 2005.

4. Para crear la base de datos se da clic derecho en la carpeta de Bases de datos que aparece en la ventana del programa.   



Y selecciona la opción Nueva base de datos...

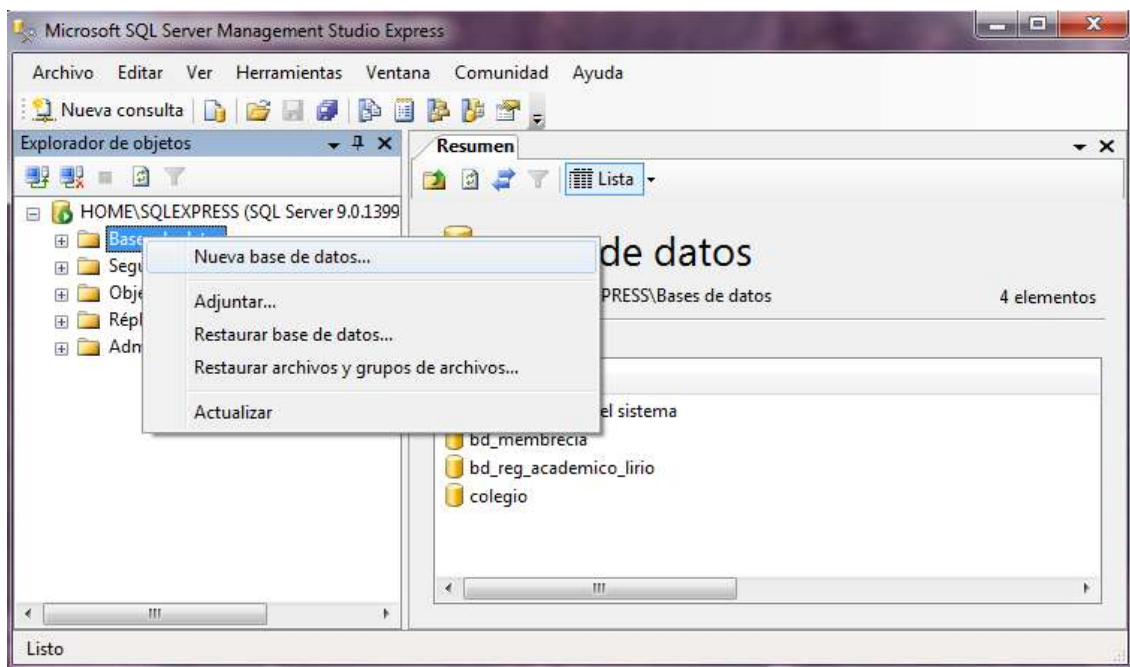


Figura No.33 Crear una Base de Datos.

- Al seleccionar la opción Nueva base de datos... se le desplegará una ventana donde ingresará el nombre de la base de dato que desea crear como se observa en la figura.

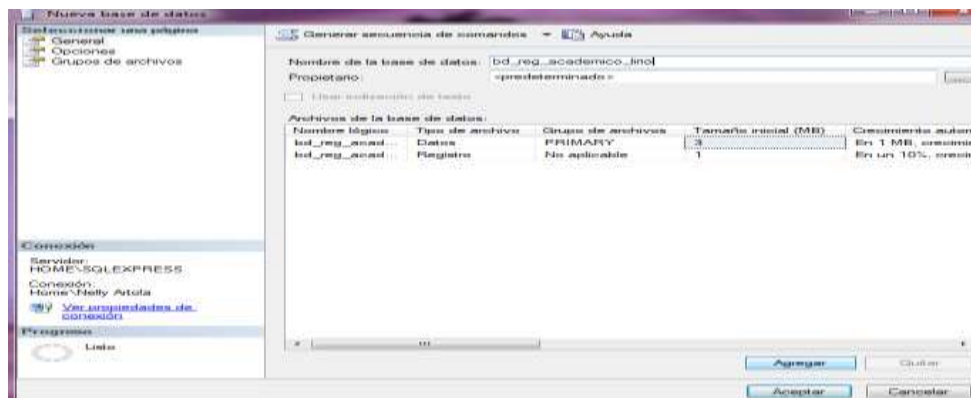


Figura No.34 Nombre a la Base de Datos.

- Para agregar los campos que tendrán las tablas de la base de datos se da clic derecho sobre la carpeta Tablas la que desplegará la sección donde podrá ingresar los campos que contendrá las tablas de su base de datos. Ver la figura.



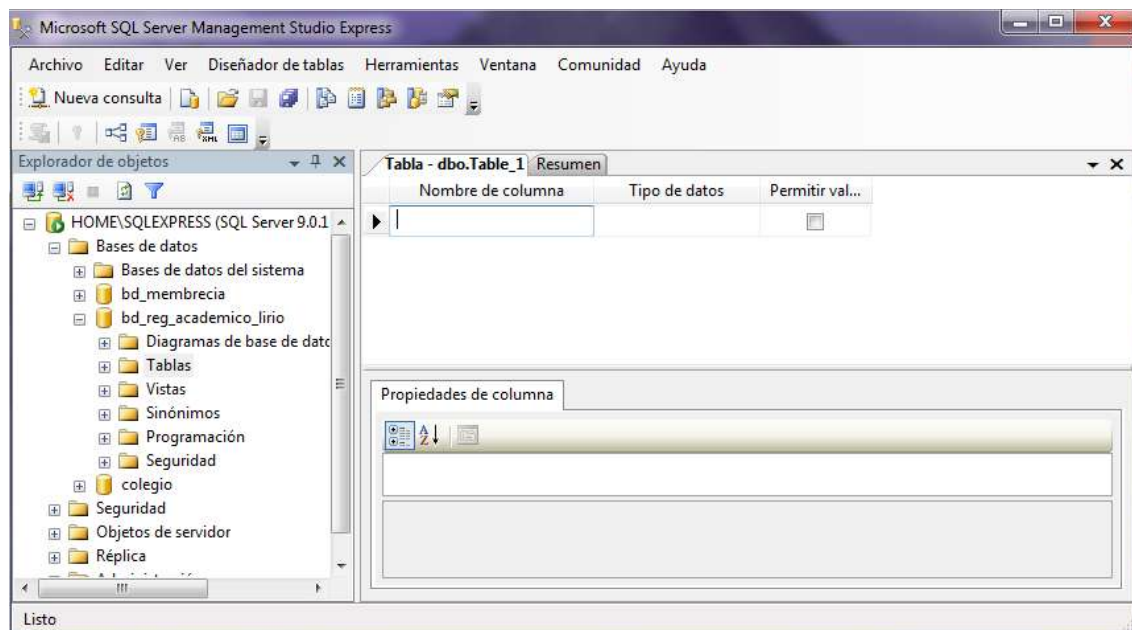


Figura No.35 Agregar Campos a una tabla de la Base de Datos.

Al tener ingresados todos los campos que posee la tabla que está creando se le dará el nombre por medio de un cuadro de dialogo que le permitirá ingresar el nombre y se le dará clic al botón aceptar. Ver la figura.

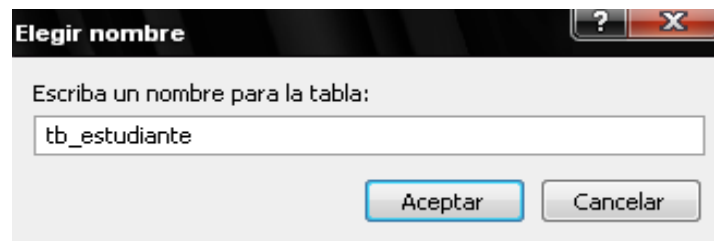


Figura No.36 Poner Nombre a una tabla de la Base de Datos.

7. Si desea modificar algún dato de las tablas de su base de datos de clic derecho sobre la tabla que desea modificar y selecciona el campo modificar tabla y luego podrá realizar la modificación. Ver la figura.

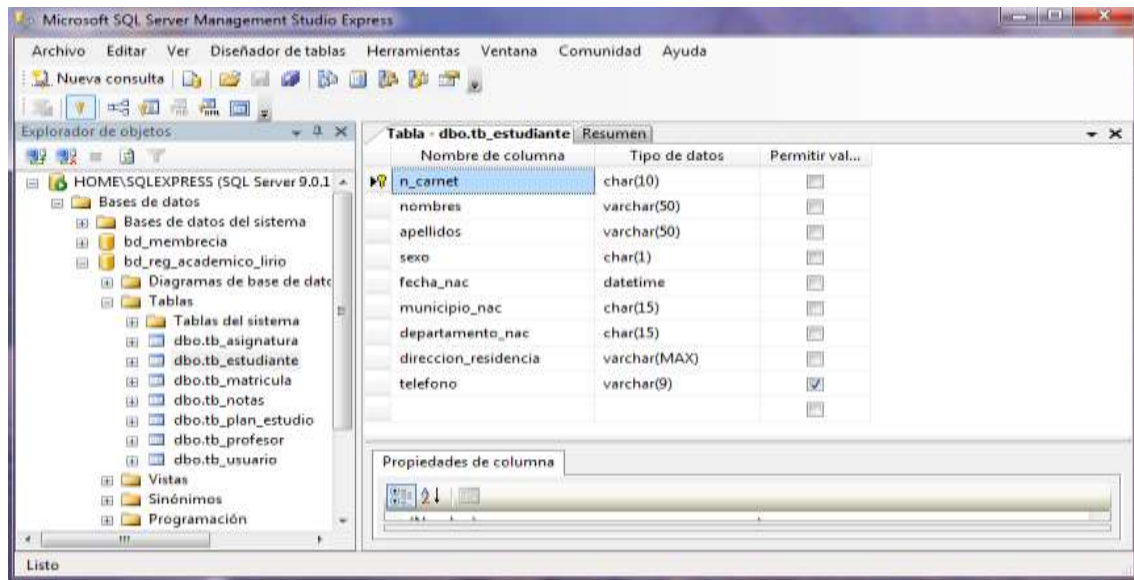


Figura No.37 Campos de una tabla de una Base de Datos.



## Códigos de ejemplo empelados en la aplicación

### Encriptar una cadena de caracteres

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.Security.Cryptography;
using System.IO;

namespace ralv
{
    public partial class frm_inicio : Form
    {
        private SqlConnection conn;
        private SqlCommand Orden;
        private SqlDataReader Lector;
        private string consulta;
        private int c = 0;
        public string rol_bd;

        //Atributos para encriptar la clave
        byte[] _key;
        byte[] _iv;

        public frm_inicio()
        {
            //Inicializamos los valores de los atributos para encriptar la clave
            _key = Encoding.ASCII.GetBytes("12EstaClave34es56dificil489ssswf");
            _iv = Encoding.ASCII.GetBytes("Devjoker7.37hAES");
            private string Desencrit(string inputText)
            {
                byte[] inputBytes = Convert.FromBase64String(inputText);
                byte[] resultBytes = new byte[inputBytes.Length];

                string textoLimpio = String.Empty;

                RijndaelManaged cripto = new RijndaelManaged();
```



```
        using (MemoryStream ms = new MemoryStream(inputBytes))
        {
            using (CryptoStream objCryptoStream = new CryptoStream(ms,
cripto.CreateDecryptor(_key, _iv),
                CryptoStreamMode.Read))
            {
                using (StreamReader sr =
                    new StreamReader(objCryptoStream, true))
                {
                    textoLimpio = sr.ReadToEnd();
                }
            }
        }

        return textoLimpio;
    }
}
```

### *Formulario MDI.*

```
using System;
using System.Collections.Generic;
using Microsoft.SqlServer.Management.Smo;
using Microsoft.SqlServer.Management.Smo.RegisteredServers;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace ralv
{
    public partial class Form1 : Form
    {
        private string rolUser;
        private string server = "localhost\\sqlexpress", dataBase = "bd_reg_academico_lirio";
        private Server srvServer = null;

        public Form1()
    }
}
```



```
{
    InitializeComponent();
}

public Form1(string strRol)
{
    InitializeComponent();
    rolUser = strRol;
}

private void Form1_Load(object sender, EventArgs e)
{
    if (rolUser == "Usuario")
    {
        nUEVOINGRESOToolStripMenuItem.Enabled = false;
        toolStripButton1.Enabled = false;

        rREGISTRARUSUARIOToolStripMenuItem.Enabled = false;
        toolStripButton6.Enabled = false;

        mMODIFICARUSUARIOToolStripMenuItem.Enabled = false;

        cOPIADESEGURIDADToolStripMenuItem.Enabled = false;
        copia_seguridad.Enabled = false;

        rESTABLECERCOPIADESEGURIDADToolStripMenuItem.Enabled = false;
        toolStripButton9.Enabled = false;
    }
}

private void nUEVOINGRESOToolStripMenuItem_Click(object sender, EventArgs e)
{
    frm_reingreso NuevoFormHijo = new frm_reingreso();
    existeFormulario(NuevoFormHijo, "No Modal");
}

private void rEINGRESOToolStripMenuItem_Click(object sender, EventArgs e)
{
    resultadobusq_est NuevoFormHijo = new resultadobusq_est();
    existeFormulario(NuevoFormHijo, "No Modal");
}

private void toolStripButton1_Click(object sender, EventArgs e)
{
    this.nUEVOINGRESOToolStripMenuItem_Click(sender, e);
}
```



```
}

private void toolStripButton2_Click(object sender, EventArgs e)
{
    this.rEINGRESOToolStripMenuItem_Click(sender, e);
}

private void pPROFESORToolStripMenuItem_Click(object sender, EventArgs e)
{
    PROFESORULTIMO NuevoFormHijo = new PROFESORULTIMO(rolUser);
    existeFormulario(NuevoFormHijo, "No Modal");
}

private void aSIGNATURAToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (tablaVacía("tb_plan_estudio"))
    {
        MessageBox.Show("PARA PODER INGRESAR ASIGNATURAS, ES NECESARIO
INGRESAR AL MENOS UN PLAN DE ESTUDIO", "ERROR", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
        return;
    }

    ASIGNATURAULTIMO NuevoFormHijo = new ASIGNATURAULTIMO(rolUser);
    existeFormulario(NuevoFormHijo, "No Modal");
}

private bool tablaVacía(string strTabla)
{
    SqlConnection conn;
    SqlCommand Orden;
    SqlDataReader Lector;
    string consulta;
    bool bRetorno = true;

    conn = new SqlConnection();
    conn.ConnectionString = "Data Source=.\sqlexpress;" + "AttachDbFilename=" +
Application.StartupPath.Substring(0, Application.StartupPath.Substring(0,
Application.StartupPath.LastIndexOf("\")).LastIndexOf("\")) +
"\\BD\\bd_reg_academico_lirio.mdf;" + "Integrated Security=True;" + "User Instance=True";

    conn.Open();
    consulta = "SELECT * FROM " + strTabla + "";

    Orden = new SqlCommand(consulta, conn);
```



```
Lector = Orden.ExecuteReader();

if (Lector.Read())
    bRetorno = false;

Lector.Close();
conn.Close();

return bRetorno;
}

private bool planActivoSinAsignaturas()
{
    SqlConnection conn;
    SqlCommand Orden;
    SqlDataReader Lector;
    string consulta;
    bool bRetorno = true;

    conn = new SqlConnection();
    conn.ConnectionString = "Data Source=.\sqlexpress;" + "AttachDbFilename=" +
Application.StartupPath.Substring(0, Application.StartupPath.Substring(0,
Application.StartupPath.LastIndexOf("\")).LastIndexOf("\") +
"\\BD\\bd_reg_academico_lirio.mdf;" + "Integrated Security=True;" + "User Instance=True";

    conn.Open();
    consulta = "SELECT tb_estudiante.n_carnet, tb_estudiante.nombres,
tb_estudiante.apellidos, tb_asignatura.nombre_asignatura, tb_notas.cod_asignatura, tb_notas.IB,
tb_notas.IIB, tb_notas.IS, tb_notas.IIIB, tb_notas.IVB, tb_notas.IIS FROM tb_asignatura
INNER JOIN tb_notas ON tb_asignatura.cod_asignatura = tb_notas.cod_asignatura INNER JOIN
tb_plan_estudio ON tb_asignatura.cod_plan_estudio = tb_plan_estudio.cod_plan_estudio INNER
JOIN tb_estudiante ON tb_estudiante.n_carnet = tb_notas.n_carnet WHERE
(tb_notas.año_lectivo = " + DateTime.Now.Year.ToString() + ") AND (tb_plan_estudio.estado =
'True')";

    Orden = new SqlCommand(consulta, conn);
    Lector = Orden.ExecuteReader();

    if (Lector.Read())
        bRetorno = false;

    Lector.Close();
    conn.Close();

    return bRetorno;
}
```



```
}

private void nOTASToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (planActivoSinAsignaturas())
    {
        MessageBox.Show("NO EXISTEN DATOS EN EL AÑO ACTUAL PARA EL PLAN
ACADÉMICO ACTUALMENTE ACTIVO", "ERROR", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        return;
    }

    NOTASULTIMO NuevoFormHijo = new NOTASULTIMO(rolUser);
    existeFormulario(NuevoFormHijo, "No Modal");
}
private void pLANDEESTUDIOToolStripMenuItem_Click(object sender, EventArgs e)
{
    frm_plan_estudio NuevoFormHijo = new frm_plan_estudio(rolUser);
    existeFormulario(NuevoFormHijo, "No Modal");
}
private void cERRARToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.Close();
}
private void toolStripButton7_Click(object sender, EventArgs e)
{
    this.Close();
}
private void toolStripButton6_Click(object sender, EventArgs e)
{
    this.rEGISTRARUSUARIOToolStripMenuItem_Click(sender, e);
}
private void toolStripButton3_Click(object sender, EventArgs e)
{
    this.pROFESORToolStripMenuItem_Click(sender, e);
}
private void toolStripButton4_Click(object sender, EventArgs e)
{
    this.aSIGNATURAToolStripMenuItem_Click(sender, e);
}
private void toolStripButton5_Click(object sender, EventArgs e)
{
    this.pLANDEESTUDIOToolStripMenuItem_Click(sender, e);
}
}
```





```
private void toolStripButton8_Click(object sender, EventArgs e)
{
    this.nOTASToolStripMenuItem_Click(sender, e);
}

private void copia_seguridad_Click(object sender, EventArgs e)
{
    this.cOPIADESEGURIDADToolStripMenuItem_Click(sender, e);
}

private void toolStripButton9_Click(object sender, EventArgs e)
{
    this.rESTABLECERCOPIADESEGURIDADToolStripMenuItem_Click(sender, e);
}

private void cASCADAToolStripMenuItem_Click(object sender, EventArgs e)
{
    LayoutMdi(MdiLayout.Cascade);
}

private void hORIZONTALToolStripMenuItem_Click(object sender, EventArgs e)
{
    LayoutMdi(MdiLayout.TileHorizontal);
}

private void vERTICALToolStripMenuItem_Click(object sender, EventArgs e)
{
    LayoutMdi(MdiLayout.TileVertical);
}

private void aCERCADEToolStripMenuItem1_Click(object sender, EventArgs e)
{
    acerca_de NuevoFormHijo = new acerca_de();
    existeFormulario(NuevoFormHijo, "No Modal");
}

private void Acerca_de_Click(object sender, EventArgs e)
{
    this.aCERCADEToolStripMenuItem1_Click(sender, e);
}

private void menuventana_DropDownOpening(object sender, EventArgs e)
{
    int cuenta = 0;

    Form[] hijos = this.MdiChildren;
    foreach (Form frm in hijos)
        cuenta++;

    if (cuenta == 0)
```



```
{
    toolStripSeparator2.Visible = false;
    cASCADAToolStripMenuItem.Enabled =
hORIZONTALToolStripMenuItem.Enabled = vERTICALToolStripMenuItem.Enabled = false;
}
else
{
    this.toolStripSeparator2.Visible = true;
    cASCADAToolStripMenuItem.Enabled =
hORIZONTALToolStripMenuItem.Enabled = vERTICALToolStripMenuItem.Enabled = true;
}
}
private void existeFormulario(Form frmArg, string strTipo)
{
    Form[] frmArray = this.MdiChildren;

    foreach (Form frm in frmArray)
    {
        if (frm.GetType() == frmArg.GetType())
        {
            frm.Focus();
            frm.WindowState = FormWindowState.Normal;
            frmArg.Dispose();
            return;
        }
    }
    if (strTipo == "Modal")
        frmArg.ShowDialog(this);
    else if (strTipo == "No Modal")
    {
        frmArg.MdiParent = this;
        frmArg.Show();
    }
    else
        MessageBox.Show("ERROR, NI MODAL, NI NO MODAL");
}
private void mATRICULAToolStripMenuItem_Click(object sender, EventArgs e)
{
    ReporteMatricula NuevoFormHijo = new ReporteMatricula();
    existeFormulario(NuevoFormHijo, "No Modal");
}
private void cONTENIDOToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show(this, "Falta poner la ayuda ...");
}
```



```

    }
    private void lISTASDEESTUDIANTESToolStripMenuItem_Click(object sender, EventArgs
e)
    {
        REPORTE_LISTA_ESTUDIANTES NuevoFormHijo = new
REPORTE_LISTA_ESTUDIANTES();
        existeFormulario(NuevoFormHijo, "No Modal");
    }
    private void bOLETINEToolStripMenuItem_Click(object sender, EventArgs e)
    {
        ReporteNotas NuevoFormHijo = new ReporteNotas();
        existeFormulario(NuevoFormHijo, "No Modal");
    }
    private void rEGISTRARUSUARIOToolStripMenuItem_Click(object sender, EventArgs e)
    {
        frm_nuevo_usuario NuevoFormHijo = new frm_nuevo_usuario(this);
        existeFormulario(NuevoFormHijo, "No Modal");
    }

    private bool soloExisteAdmin()
    {
        SqlConnection conn;
        SqlCommand Orden;
        SqlDataReader Lector;
        string consulta;
        bool bRetorno = true;

        conn = new SqlConnection();
        conn.ConnectionString = "Data Source=.\sqlexpress;" + "AttachDbFilename=" +
Application.StartupPath.Substring(0, Application.StartupPath.Substring(0,
Application.StartupPath.LastIndexOf("\").LastIndexOf("\") +
"\\BD\\bd_reg_academico_lirio.mdf;" + "Integrated Security=True;" + "User Instance=True";

        conn.Open();
        consulta = "SELECT * FROM tb_usuario WHERE usuarios != 'Administrador'";

        Orden = new SqlCommand(consulta, conn);
        Lector = Orden.ExecuteReader();

        if (Lector.Read())
            bRetorno = false;

        Lector.Close();
        conn.Close();
    }

```



```

        return bRetorno;
    }

    private void mODIFICARUSUARIOToolStripMenuItem_Click(object sender, EventArgs e)
    {
        if (soloExisteAdmin())
        {
            MessageBox.Show("ACTUALMENTE EL SISTEMA SOLO TIENE REGISTRADO
AL USUARIO ADMINISTRADOR POR DEFECTO " + "\n" + "INGRESE UN NUEVO
USUARIO DENTRO DEL SISTEMA", "ERROR", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
            return;
        }

        Modificar_Usuario NuevoFormHijo = new Modificar_Usuario();
        existeFormulario(NuevoFormHijo, "No Modal");
    }
    private void toolStripButton10_Click(object sender, EventArgs e)
    {
        this.mODIFICARUSUARIOToolStripMenuItem_Click(sender, e);
    }
}

```

*Expresión regular para quitar múltiples espacios en blanco.*

```

private string quitarMultiplesEspaciosEnBlanco(string args)
{
    return Regex.Replace(args.Trim(), @"\s+", " ");
}

```

*Reporte Listado de Estudiantes.*

```

using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using CrystalDecisions.Shared;
using CrystalDecisions.ReportSource;
using CrystalDecisions.CrystalReports.Engine;

```



```

using System.Data.SqlClient;
using ralv.Reportes;

namespace ralv.Mostrar_Reportes
{
    public partial class ListadoEstudiantes : Form
    {
        private SqlConnection conn;
        private string grado;
        private string turno;
        private int año;
        private REPORTE_LISTA_ESTUDIANTES rleAnterior;

        public ListadoEstudiantes(string grad, string tur, int an,
REPORTE_LISTA_ESTUDIANTES rle)
        {
            conn = new SqlConnection();
            conn.ConnectionString = "Data Source=.\sqlexpress;" + "AttachDbFilename=" +
Application.StartupPath.Substring(0, Application.StartupPath.Substring(0,
Application.StartupPath.LastIndexOf("\")).LastIndexOf("\")) +
"\BD\bd_reg_academico_lirio.mdf;" + "Integrated Security=True;" + "User Instance=True";
            InitializeComponent();
            grado = grad;
            turno = tur;
            año = an;
            rleAnterior = rle;
        }

        private void crystalReportViewer1_Load(object sender, EventArgs e)
        {
            conn.Open();
            CrystalReportEstudiante opt = new CrystalReportEstudiante();
            DataSet objDataSet = new DataSet();
            SqlDataAdapter objAdapt = new SqlDataAdapter("SELECT dbo.tb_estudiante.n_carnet,
dbo.tb_estudiante.nombres, dbo.tb_estudiante.apellidos, dbo.tb_matricula.grado_cursa,
dbo.tb_matricula.modalidad, dbo.tb_matricula.año_lectivo FROM  dbo.tb_estudiante INNER
JOIN dbo.tb_matricula ON dbo.tb_estudiante.n_carnet = dbo.tb_matricula.n_carnet WHERE
grado_cursa = " + grado + " and tb_matricula.modalidad = " + turno + " and
tb_matricula.año_lectivo = " + año + "ORDER BY n_carnet ASC", conn);
            SqlDataAdapter objMatricu = new SqlDataAdapter("SELECT dbo.tb_estudiante.n_carnet,
dbo.tb_estudiante.nombres, dbo.tb_estudiante.apellidos, dbo.tb_matricula.grado_cursa,
dbo.tb_matricula.modalidad, dbo.tb_matricula.año_lectivo FROM  dbo.tb_estudiante INNER
JOIN dbo.tb_matricula ON dbo.tb_estudiante.n_carnet = dbo.tb_matricula.n_carnet WHERE
grado_cursa = " + grado + " and tb_matricula.modalidad = " + turno + " and
tb_matricula.año_lectivo = " + año + "ORDER BY n_carnet ASC", conn);

```



```
objAdapt.Fill(objDataSet, "tb_estudiante");  
objMatricu.Fill(objDataSet, "tb_matricula");  
opt.SetDataSource(objDataSet);  
crystalReportViewer1.ReportSource = opt;  
conn.Close();  
  
}  
  
}
```